
Научно-исследовательский институт
«Центрпрограммсистем»

Программные продукты и системы

МЕЖДУНАРОДНЫЙ НАУЧНО-ПРАКТИЧЕСКИЙ ЖУРНАЛ

№ 4 (112), 2015

Главный редактор
С.В. ЕМЕЛЬЯНОВ, академик РАН

Тверь

PROGRAMMNYE PRODUKTY I SISTEMY (SOFTWARE & SYSTEMS)

International research and practice journal

no. 4 (112), 2015

Editor-in-Chief
S.V. EMELYANOV, Academician of the Russian Academy of Sciences

Tver
Russian Federation

Research Institute CENTERPROGRAMSYSTEM

© ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

Международное научно-практическое
приложение к международному журналу
«ПРОБЛЕМЫ ТЕОРИИ И ПРАКТИКИ УПРАВЛЕНИЯ»

Главный редактор

С.В. ЕМЕЛЬЯНОВ, академик РАН (г. Москва, Россия)

Научные редакторы:

В.Н. РЕШЕТНИКОВ, д.ф.-м.н., профессор МАТИ (г. Москва, Россия)

Н.А. СЕМЕНОВ, д.т.н., профессор ТвГТУ (г. Тверь, Россия)

В.Е. ШУКШУНОВ, д.т.н., профессор, Центр тренажеростроения (г. Москва, Россия)

Рецензенты:

А.П. Еремеев, д.т.н., профессор МЭИ (ТУ) (г. Москва, Россия)

В.Н. Решетников, д.ф.-м.н., профессор МАТИ (г. Москва, Россия)

Н.А. Семенов, д.т.н., профессор ТвГТУ (г. Тверь, Россия)

О.А. Финько, д.т.н., профессор ФВАС (г. Краснодар, Россия)

А.Н. Сотников, д.ф.-м.н., профессор МСЦ РАН (г. Москва, Россия)

Т.Н. Гартман, д.т.н., профессор РХТУ им. Д.И. Менделеева (г. Москва, Россия)

А.В. Язенин, д.ф.-м.н., профессор ТвГУ (г. Тверь, Россия)

Издатель **НИИ «Центрпрограммсистем»**
(г. Тверь, Россия)

Учредители: МНИИПУ (г. Москва, Россия),
Главная редакция международного журнала
«Проблемы теории и практики управления»
(г. Москва, Россия),

Закрытое акционерное общество
«Научно-исследовательский институт
«Центрпрограммсистем» (г. Тверь, Россия)

*Журнал зарегистрирован
в Комитете Российской Федерации
по печати 26 июня 1995 г.*

*Регистрационное
свидетельство № 013831*

Подписной индекс в каталоге

Агентства «Роспечать» 70799

ISSN 0236-235X (печатн.)

ISSN 2311-2735 (онлайн)

МЕЖДУНАРОДНАЯ РЕДАКЦИОННАЯ КОЛЛЕГИЯ

Семенов Н.А. – д.т.н., профессор Тверского государственного технического университета, заместитель главного редактора (г. Тверь, Россия)

Решетников В.Н. – д.ф.-м.н., профессор Российского государственного технологического университета им. К.Э. Циолковского (МАТИ), заместитель главного редактора (г. Москва, Россия)

Арефьев И.Б. – д.т.н., профессор Морской академии Польши (г. Щецин, Польша)

Афанасьев А.П. – д.ф.-м.н., профессор Московского физико-технического института (технического университета), заведующий Центром распределенных вычислений Института проблем передачи информации РАН (г. Москва, Россия)

Батыршин И.З. – д.т.н., профессор Мексиканского института нефти (г. Мехико, Мексика)

Вагин В.Н. – д.т.н., профессор Московского энергетического института (технического университета) (г. Москва, Россия)

Голенков В.В. – д.т.н., профессор Белорусского государственного университета информатики и радиоэлектроники (г. Минск, Беларусь)

Еремеев А.П. – д.т.н., профессор Московского энергетического института (технического университета) (г. Москва, Россия)

Котов А.С. – кандидат наук, ассистент профессора университета Уэйна (штат Мичиган) (г. Детройт, США)

Кузнецов О.П. – д.т.н., профессор Института проблем управления РАН (г. Москва, Россия)

Курейчик В.М. – д.т.н., профессор Технологического института Южного федерального университета (г. Таганрог, Россия)

Лисецкий Ю.М. – к.т.н., генеральный директор «S&T Ukraine» (г. Киев, Украина)

Мамросенко К.А. – к.т.н., доцент Российского государственного технологического университета им. К.Э. Циолковского (МАТИ), заведующий отделом Центра визуализации и спутниковых информационных технологий ФНЦ НИИСИ РАН (г. Москва, Россия)

Нгуен Тхань Нгу – д.ф.-м.н., профессор, проректор Ханойского открытого университета (г. Ханой, Вьетнам)

Николов Р.В. – доктор наук, профессор Университета библиотековедения и информационных технологий Софии (г. София, Болгария)

Осипов Г.С. – д.ф.-м.н., профессор, заместитель директора Института системного анализа РАН (г. Москва, Россия)

Палюх Б.В. – д.т.н., профессор Тверского государственного технического университета (г. Тверь, Россия)

Рахманов А.А. – д.т.н., профессор, заместитель генерального директора Концерна «РТИ Системы» (г. Москва, Россия)

Серов В.С. – д.ф.-м.н., профессор Университета прикладных наук Оулу (г. Оулу, Финляндия)

Сотников А.Н. – д.ф.-м.н., профессор, Межведомственный суперкомпьютерный центр РАН (г. Москва, Россия)

Сулейманов Д.Ш. – академик АН Республики Татарстан, д.т.н., профессор Казанского государственного технического университета (г. Казань, Республика Татарстан, Россия)

Тарасов В.Б. – к.т.н., доцент Московского государственного технического университета им. Н.Э. Баумана (г. Москва, Россия)

Таратухин В.В. – доктор философии, управляющий директор Европейского исследовательского центра в области информационных систем (ERCIS) Вестфальского университета им. Вильгельма (г. Мюнстер, Германия)

Хорошевский В.Ф. – д.т.н., профессор Московского физико-технического института (технического университета) (г. Москва, Россия)

Язенин А.В. – д.ф.-м.н., профессор Тверского государственного университета (г. Тверь, Россия)

АССОЦИИРОВАННЫЕ ЧЛЕНЫ РЕДАКЦИИ

Московский энергетический институт (технический университет), г. Москва, Россия

Технологический институт Южного федерального университета, г. Таганрог, Россия

Тверской государственный технический университет, г. Тверь, Россия

Научно-исследовательский институт «Центрпрограммсистем», г. Тверь, Россия

АДРЕС РЕДАКЦИИ

Россия, 170024, г. Тверь, пр. 50 лет Октября, 3а

Телефон (482-2) 39-91-49

Факс (482-2) 39-91-00

E-mail: red@cps.tver.ru

www.swsys.ru

Подписано в печать 17.11.2015 г.

Отпечатано ООО ИПП «Фактор и К»

Россия, 170028, г. Тверь, ул. Лукина, д. 4, стр. 1

Выпускается один раз в квартал. Год издания двадцать восьмой

Формат 60×84 1/8. Объем 264 стр.

Заказ № 68. Тираж 1000 экз. Цена 257,40 руб.

Автор статьи отвечает за подбор, оригинальность и точность приводимого фактического материала.

Авторские гонорары не выплачиваются. При перепечатке материалов ссылка на журнал обязательна.

© PROGRAMMNYE PRODUKTY I SISTEMY
(SOFTWARE & SYSTEMS)

International research and practice supplement for International magazine
"THEORETICAL AND PRACTICAL ISSUES OF MANAGEMENT"

Editor-in-chief

S.V. Emelyanov, Academician of the Russian Academy of Sciences (Moscow, Russian Federation)

Science editors:

V.N. Reshetnikov, Dr.Sc. (Physics and Mathematics), Professor MATI (Moscow, Russian Federation)

N.A. Semenov, Dr.Sc. (Engineering), Professor TSTU (Tver, Russian Federation)

V.E. Shukshunov, Dr.Sc. (Engineering), Professor, Space Simulator Center (Moscow, Russian Federation)

Reviewers:

A.P. Ereemeev, Dr.Sc. (Engineering), Professor MPEI (Moscow, Russian Federation)

V.N. Reshetnikov, Dr.Sc. (Physics and Mathematics), Professor MATI (Moscow, Russian Federation)

N.A. Semenov, Dr.Sc. (Engineering), Professor TSTU (Tver, Russian Federation)

O.A. Finko, Dr.Sc. (Engineering), Professor Military Academy of Communications (Krasnodar, Russian Federation)

A.N. Sotnikov, Dr.Sc. (Physics and Mathematics), Professor JSCC RAS (Moscow, Russian Federation)

T.N. Gartman, Dr.Sc. (Engineering), Professor D. Mendeleev University (Moscow, Russian Federation)

A.V. Yazenin, Dr.Sc. (Physics and Mathematics), Professor TSU (Tver, Russian Federation)

**Publisher Research Institute
CENTERPROGRAMSYSTEM
(Tver, Russian Federation)**

The Founders: International Scientific
and Research Institute
for Management Issues
(Moscow, Russian Federation),
the Chief Editorial Board
of International Magazine "Theoretical
and practical issues of management"
(Moscow, Russian Federation),
Research Institute
CENTERPROGRAMSYSTEM
(Tver, Russian Federation)
The magazine is on record
in Russian committee
on press 26th of June 1995
Registration certificate № 013831
ISSN 0236-235X (print)
ISSN 2311-2735 (online)

INTERNATIONAL EDITORIAL BOARD

Semenov N.A. – Dr.Sc. (Engineering), Professor of Tver State Technical University, Deputy Editor-in-Chief
(Tver, Russian Federation)

Reshetnikov V.N. – Dr.Sc. (Physics and Mathematics), Professor of Russian State Technological University (MATI),
Deputy Editor-in-Chief (Moscow, Russian Federation)

Arefev I.B. – Dr.Sc. (Engineering), Professor of Poland Szczecin Maritime Academy (Szczecin, Poland)

Afanas'ev A.P. – Dr.Sc. (Physics and Mathematics), Professor of Moscow Institute of Physics and Technology,
Head of Centre for Distributed Computing of Institute for Information Transmission Problems (Moscow, Russian Federation)

Batyrshin I.Z. – Dr.Sc. (Engineering), Professor of Mexican Petroleum Institute (Mexico City, Mexico)

Vagin V.N. – Dr.Sc. (Engineering), Professor of Moscow Power Engineering Institute (Technical University)
(Moscow, Russian Federation)

Golenkov V.V. – Dr.Sc. (Engineering), Professor of Belarusian State University of Informatics and Radioelectronics
(Minsk, Republic of Belarus)

Ereemeev A.P. – Dr.Sc. (Engineering), Professor of Moscow Power Engineering Institute (Technical University)
(Moscow, Russian Federation)

Kotov A.S. – Ph.D. (Computer Science), Assistant Professor, Wayne State University (Detroit, MI, USA)

Kuznetsov O.P. – Dr.Sc. (Engineering), Professor of the Institute of Control Sciences of the Russian Academy of Sciences
(Moscow, Russian Federation)

Kureichik V.M. – Dr.Sc. (Engineering), Professor of Taganrog Technology Institute at Southern Federal University
(Taganrog, Russian Federation)

Lisetskiy Yu.M. – Ph.D.Tech.Sc., CEO of "S&T Ukraine" (Kiev, Ukraine)

Mamrosenko K.A. – Ph.D. (Engineering), Associate Professor of Russian State Technological University (MATI),
Head of Department of Center of visualization and satellite information technologies SRISA RAS (Moscow, Russian Federation)

Nguyen Thanh Nghi – Dr.Sc. (Physics and Mathematics), Professor, Vice-Principal of Hanoi Open University (Hanoi, Vietnam)

Nikolov R.V. – Full Professor of the University of Library Studies and Information Technology (Sofia, Bulgaria)

Osipov G.S. – Dr.Sc. (Physics and Mathematics), Professor, Deputy of the Principal of Institute of Systems Analysis
of the Russian Academy of Sciences (Moscow, Russian Federation)

Palyukh B.V. – Dr.Sc. (Engineering), Professor of Tver State Technical University (Tver, Russian Federation)

Rakhmanov A.A. – Dr.Sc. (Engineering), Professor, Deputy of the CEO of Concern "RTI Systems"
(Moscow, Russian Federation)

Serov V.S. – Dr.Sc. (Physics and Mathematics), Professor of the Oulu University of Applied Sciences (Oulu, Finland)

Sotnikov A.N. – Dr.Sc. (Physics and Mathematics), Professor, Joint Supercomputer Center of the Russian Academy
of Sciences (Moscow, Russian Federation)

Suleimanov D.Sh. – Academician of TAS, Dr.Sc. (Engineering), Professor of Kazan State Technical University
(Kazan, Republic of Tatarstan, Russian Federation)

Tarassov V.B. – Ph.D. (Engineering), Associate Professor of Bauman Moscow State Technical University
(Moscow, Russian Federation)

Taratoukhine V.V. – Ph.D. (Engineering), Dr.Ph., Managing Director of the Competence Centre ERP and ERCIS Lab
Russia of the ERCIS (Muenster, Germany)

Khoroshevsky V.F. – Dr.Sc. (Engineering), Professor of Moscow Institute of Physics and Technology
(Moscow, Russian Federation)

Yazenin A.V. – Dr.Sc. (Physics and Mathematics), Professor of Tver State University (Tver, Russian Federation)

ASSOCIATED EDITORIAL BOARD MEMBERS

Moscow Power Engineering Institute (Technical University), Moscow, Russian Federation

Technology Institute at Southern Federal University, Taganrog, Russian Federation

Tver State Technical University, Tver, Russian Federation

Research Institute CENTERPROGRAMSYSTEM, Tver, Russian Federation

EDITORIAL OFFICE ADDRESS

50 let Oktyabrya Ave. 3a, Tver, 170024, Russian Federation

Phone: (482-2) 39-91-49 Fax: (482-2) 39-91-00

E-mail: red@cps.tver.ru

www.swsys.ru

Passed for printing 17.11.2015

Printed in printing-office "Faktor i K"

Lukina St. 4/1, Tver, 170028, Russian Federation

Published quarterly. 28th year of publication

Format 60×84 1/8. Circulation 1000 copies

Prod. order № 68. Wordage 264 pages. Price 257,40 rub.

Вниманию авторов!

Международный журнал «Программные продукты и системы» публикует материалы научного и научно-практического характера по новым информационным технологиям, результаты академических и отраслевых исследований в области использования средств вычислительной техники. Практикуются выпуски тематических номеров по искусственному интеллекту, системам автоматизированного проектирования, по технологиям разработки программных средств и системам защиты, а также специализированные выпуски, посвященные научным исследованиям и разработкам отдельных вузов, НИИ, научных организаций.

Решением Президиума Высшей аттестационной комиссии (ВАК) Министерства образования и науки РФ международный журнал «Программные продукты и системы» внесен в Перечень ведущих рецензируемых научных журналов и изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученых степеней кандидата и доктора наук.

Информация об опубликованных статьях по установленной форме регулярно предоставляется в систему Российского индекса научного цитирования (РИНЦ), в CrossRef и готовится для передачи в международные базы цитирования.

Условия публикации

К рассмотрению принимаются ранее нигде не опубликованные материалы, соответствующие тематике журнала (специализация 05.13.XX – Информатика, вычислительная техника и управление) и отвечающие редакционным требованиям.

Работа представляется в электронном виде в формате Word. При обилии сложных формул обязательно наличие статьи и в формате PDF. Формулы должны быть набраны в редакторе формул Word (Microsoft Equation или MathType). Объем статьи вместе с иллюстрациями – не менее 10 000 знаков. Диаграммы, схемы, графики должны быть доступными для редактирования (Word, Visio, Excel). Все иллюстрации для полиграфического воспроизведения представляются в черно-белом варианте. Цветные, тонированные, отсканированные, не подлежащие редактированию средствами Word рисунки и экранные формы следует присылать в хорошем качестве для их дополнительного размещения на сайте журнала в макете статьи с доступом по ссылке. (Публикация материалов с использованием гипертекста, графики, аудио-, видео-, программных средств и др. возможна в электронном издании «Программные продукты, системы и алгоритмы», сайт www.swsys-web.ru.) Заголовок должен быть информативным; сокращения, а также терминологию узкой тематики желательно в нем не использовать. Количество авторов на одну статью – не более 4, количество статей одного автора в номере, включая соавторство, – не более 2. Список литературы (оформленный в соответствии с ГОСТ Р 7.05–2008), наличие которого обязательно, должен включать не менее 10 пунктов.

Необходимы также аннотация (не менее 200 слов), ключевые слова (7–10) и индекс УДК. Название статьи, аннотация и ключевые слова должны быть переведены на английский язык (машинный перевод недопустим), а фамилии авторов, названия и юридические адреса организаций (если нет официального перевода), пристайные списки литературы – транслитерированы по стандарту BGN/PCGN.

Вместе со статьей следует прислать отзыв-рекомендацию в произвольной форме, экспертное заключение, лицензионное соглашение, а также сведения об авторах: фамилия, имя, отчество, название и юридический адрес организации, должность, ученые степень и звание (если есть), контактный телефон, электронный адрес, почтовый адрес для отправки бесплатного авторского экземпляра журнала.

Порядок рецензирования

Все статьи, поступающие в редакцию (соответствующие тематике и оформленные согласно требованиям к публикации), подлежат обязательному рецензированию в течение месяца с момента поступления.

В редакции есть устоявшийся коллектив рецензентов, среди которых члены международной редколлегии журнала, эксперты из числа крупных специалистов в области информатики и вычислительной техники ведущих вузов страны, а также ученые и специалисты НИИ «Центрпрограммсистем» (г. Тверь).

Рецензирование проводится конфиденциально. Автору статьи предоставляется возможность ознакомиться с текстом рецензии. При необходимости статья отправляется на доработку.

Рецензии обсуждаются на заседаниях рабочей группы, состоящей из членов научного совета журнала. Заседания проводятся раз в месяц в НИИ «Центрпрограммсистем» (г. Тверь), где принимается решение о целесообразности публикации статьи.

Статьи, одобренные редакционным советом, публикуются бесплатно в течение года с момента одобрения, а отправленные на доработку – с момента поступления после устранения замечаний.

Редакция международного журнала «Программные продукты и системы» в своей работе руководствуется сводом правил Кодекса этики научных публикаций, разработанным и утвержденным Комитетом по этике научных публикаций.

УДК 004.72

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.005-015

КОНЦЕПТУАЛЬНЫЕ ОСНОВЫ РАЗРАБОТКИ И СОЗДАНИЯ УЧЕБНО-ТРЕНАЖЕРНО-МОДЕЛИРУЮЩЕГО КОМПЛЕКСА НОВОГО ПОКОЛЕНИЯ

В.Е. Шукишунов, д.т.н., профессор, генеральный директор, v.shukshunov@simct.ru;

В.В. Янюшкин, к.т.н., зам. генерального директора, v.yanyushkin@simct.ru

(Центр тренажеростроения и подготовки персонала,

ул. Первомайская, 92, г. Москва, 115088, Россия)

В статье рассматриваются подходы к созданию учебно-тренажерно-моделирующего комплекса нового поколения в НИИ ЦПК имени Ю.А. Гагарина на базе современных технических и программных решений. Приведены подходы к декомпозиции задач проектирования учебно-тренажерно-моделирующего комплекса нового поколения на основе нескольких взаимосвязанных этапов решения определенных задач. Дается краткий анализ создания и эксплуатации нескольких поколений тренажерных комплексов для подготовки космонавтов, в том числе с точки зрения использования в них интегрирующих принципов. Предлагается алгоритм интеграции существующих и эксплуатируемых технических средств подготовки космонавтов в единый тренажерный комплекс, при этом рассматриваются понятие «шлюз» и современные технологии вычислений, в том числе применение виртуализации и набора специализированных технических и программных средств. Приводится перспективная структура учебно-тренажерно-моделирующих комплексов, каждый из которых представляет собой отдельную составную часть, решающую определенный круг задач, особое внимание уделяется «сращиванию» образовательного и тренажерного процессов при подготовке специалистов. В заключение приводится состав экспериментального стенда, на базе которого апробировались все предлагаемые решения по созданию учебно-тренажерно-моделирующего комплекса. Описываются полученные результаты, которые отражают принципиально новые решения, в том числе применение централизованных расчетов единой вычислительной системы, применение на рабочих местах тонких клиентов. В процессе работы анализируются загрузка ресурсов, сетевых каналов передачи данных и других критических параметров. Отмечается высокая оценка многими экспертами в области информационных технологий результатов проведенных работ для создания экспериментального стенда.

Ключевые слова: учебно-тренажерно-моделирующий комплекс, технологии виртуализации ресурсов, технические средства подготовки космонавтов, интеграция автономных тренажеров, единая программно-аппаратная инфраструктура.

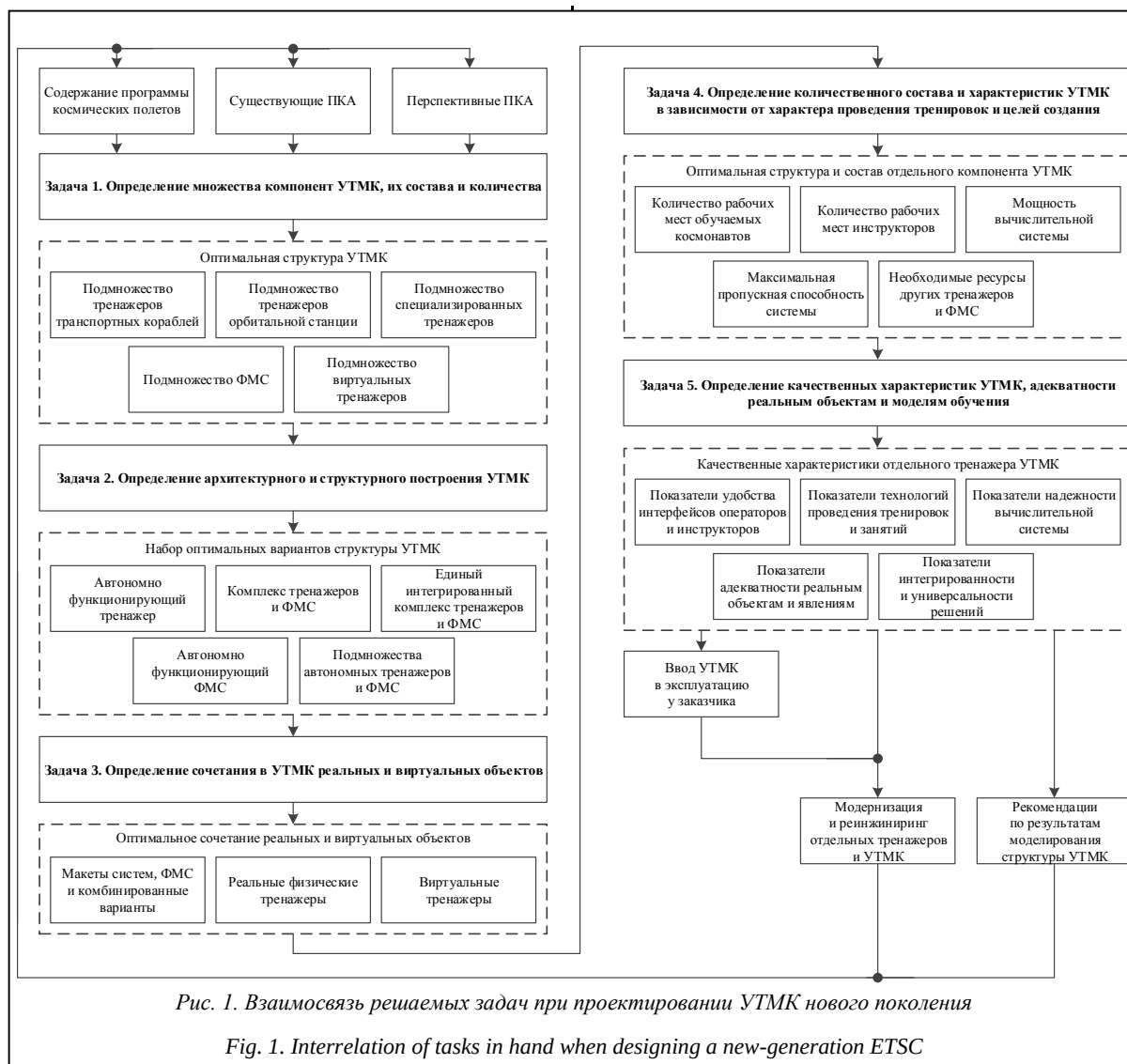
При разработке новых тренажеров, функционально-моделирующих стендов (ФМС) и специализированных комплексов в рамках единой инфраструктуры учебно-тренажерной базы необходимо создать теоретические основы проектирования. Для этого может быть использована декомпозиция этапов проектирования с целью упрощения последующего использования, модернизации и развития.

На рисунке 1 показана взаимосвязь основных задач в процессе проектирования учебно-тренажерно-моделирующего комплекса (УТМК) нового поколения, при этом каждая из них представляет собой отдельную теоретическую проблему в комплексе, составляющую один из этапов разработки. Исходными данными для проектирования являются программы космических полетов, а также существующие и перспективные пилотируемые космические аппараты (ПКА). После формулировки каждой задачи показаны результаты оптимальных выбранных структур и показателей этапа проектирования. Как видно из рисунка, даже после окончания разработки и ввода в эксплуатацию УТМК возможна его дальнейшая неоднократная модернизация в соответствии с введенными новыми задачами и требованиями без разрушения существующих технологических и программных решений. Одним из вариантов является создание имитационной модели полного цикла проектирования и разработки

сложного распределенного тренажера, позволяющей оценить ожидаемые технико-экономические показатели и эффективность решений до их непосредственной реализации у заказчика с выдачей рекомендаций и корректировкой исходных данных и целей создания УТМК.

Содержание программ космических полетов, типы и количество участвующих в их реализации ПКА определяют, какое количество и типы тренажеров и систем обучения необходимы для реализации этих программ. Оптимальную структуру УТМК целесообразно представить в виде набора подмножеств: подмножества тренажеров транспортных космических кораблей, подмножества тренажеров международных космических станций (МКС), подмножества специализированных тренажеров, ФМС и систем виртуальной реальности. Данные подмножества определяют состав тренажерного, научного и образовательного модулей тренажера нового поколения [1].

Далее решается задача анализа возможных вариантов структурного построения УТМК, когда комплекс тренажеров представляет собой простой набор автономно функционирующих тренажеров, единый интегрированный тренажерный комплекс, формируемый из типовых программно-аппаратных модулей, имеющих стандартный интерфейс. Возможна комбинация структуры построения, сочетающей в себе как локальные тренажеры, так и еди-



ные тренажерные комплексы, а также объединение автономных тренажеров в самостоятельное подмножество, которое интегрируется с помощью локальной вычислительной сети (ЛВС) с другими УТМК. На данном этапе по сути рассматриваются варианты системы интеграции модулей (или ее наличие), которая будет являться центральным, объединяющим потоки информации составным компонентом. При использовании единого распределенного информационного пространства, интегрирующего существующие учебную и тренажерную базы, система интеграции модулей каждого тренажера связана набором интерфейсов с составными частями общего информационного пространства с помощью специализированного ПО.

Задача определения сочетания в УТМК реальных и виртуальных объектов, с одной стороны, решает вопросы минимизации финансовых затрат на создание УТМК, а с другой – максимизирует критерий полноты подготовки экипажей космонавтов к полному объему выполнения космической программы, особенно в части нештатных и аварийных

ситуаций, которые трудно реализуются на физических тренажерах, но возможны на виртуальных.

Задачи определения количественного состава и характеристик УТМК в зависимости от характера проведения тренировок и целей создания определяют такие параметры, как оптимальный поток тренировок, обуславливающий количество космонавтов, одновременно использующих программно-технические средства УТМК, обеспечение максимальной пропускной способности, количество рабочих мест инструкторов, мощность и характеристики вычислительных ресурсов для обработки информации и пропускная способность вычислительной системы, а также ЛВС. При проектировании эти требования определяют состав ресурсов, выделяемых из общего центра обработки данных (ЦОД) учебно-тренажерной базы, или ресурсов, которые необходимо добавить в ЦОД с последующим выделением виртуальных серверных мощностей для вычисления, обработки данных, хранения БД. Также, исходя из этого, определяются нагрузка на каналы связи и возможности сетевой подсистемы единой учебно-тренажерной

базы с учетом внешних подключений по проводным и беспроводным каналам связи множества пользователей.

Определение качественных характеристик УТМК, адекватности реальным объектам и моделям обучения, в том числе по критериям интерактивности тренировочного процесса, дистанционного доступа к ресурсам, максимальной адекватности реальному объекту, современным технологиям проведения тренировок, адаптивности тренировочного процесса в зависимости от результатов оперативного тестирования и оценки действий космонавта в процессе тренировки, а также общей надежности комплекса

с точки зрения программных и технических средств отражается уровнем проектирования модуля управления тренировками и занятиями, а также информационными и обеспечивающими системами тренажера нового поколения.

Поэтапная проработка каждой из перечисленных задач с выходными результатами, связывающимися в общую идеологию состава модулей УТМК нового поколения, аппаратно и программно интегрированного в единый ЦОД учебно-тренажерной базы ЦПК им. Ю.А. Гагарина (далее ЦПК), позволит получить высокую эффективность использования проектируемых тренажеров в подготовке космонавтов.

Для определения конкретных технических предложений для проектирования УТМК нового поколения на базе имеющихся учебной и тренажерной баз ЦПК необходимо провести анализ существующих в проектировании и разработке тренажерных комплексов подготовки космонавтов подходов, а также выявить особенности развития и эксплуатации комплексов.

Исторические аспекты проектирования нескольких поколений тренажерных комплексов

Относительно технических и учебно-методических средств подготовки космонавтов ЦПК никогда не отставал от требований времени, от зарубеж-



Рис. 2. Типовой состав подсистем индивидуального и коллективного пользования космического тренажера

Fig. 2. A type composition of space simulator private and multiple-access subsystems

ных средств подготовки astronauts. Все *технические средства подготовки космонавтов* (ТСПК) были построены на базе современных достижений науки, новых принципов и эффективных технологий, при этом они и сейчас не уступают ТСПК зарубежных astronauts [2, 3].

Однако требования к содержанию и объему средств подготовки космонавтов постоянно растут. К тому же нельзя не принимать во внимание появление новых средств обработки информации и информационных технологий.

Более 35 лет назад вышло специальное Постановление Совета министров СССР о создании *тренажерно-моделирующего комплекса* (ТМК). Данная задача усилиями коллективов ЦПК (ныне – ФГБУ «НИИ ЦПК им. Ю.А. Гагарина») и ОКБ «Орбита» (сегодня – ООО «Центр тренажеростроения и подготовки персонала») – головной организации по созданию ТМК – была решена в течение 1980–1986 гг. [3, 4]. Основная идея его построения состояла в том, что весь состав средств тренажера разделялся на два класса (рис. 2): средства коллективного пользования, которые можно унифицировать и использовать в разных тренажерах при проведении тренировок экипажей космонавтов, и средства индивидуального пользования, жестко связанные с одним конкретным тренажером.

Построение тренажерной системы, ориентированное на интеграцию тренажерных средств (систем коллективного пользования), целесообразно с точки

зрения как экономии средств, так и возможностей комплекса: количество средств коллективного пользования можно уменьшить в несколько раз, ограничивая их числом одновременно работающих тренажеров; при технических отказах и неисправностях можно использовать другой аналогичный ресурс; уменьшение объема технических средств влечет за собой сокращение численности эксплуатационного персонала; появляется возможность реализации режима совместной работы тренажеров и взаимодействия их экипажей при отработке задач, требующих координации [5].

Данные принципы были использованы начиная с тренажерного комплекса «Белладонна», который был разработан и создан ОКБ «Орбита» в начале 80-х годов XX века для построения на его базе тренажеров для подготовки космонавтов по программе «Долговременная орбитальная станция «Салют». В нем функциональные блоки интегрировались в единый комплекс средствами *вычислительной системы* (ВС). Архитектура тренажерного комплекса «Белладонна» в силу модульного построения позволяла проводить непрерывную модернизацию и наращивание количества тренажеров в его составе путем добавления лишь терминальных комплексов *устройств связи с объектами* (УСО) для новых рабочих мест [3].

Топология структуры тренажерного комплекса «Ермак-27» (орбитальный комплекс «Мир») практически не изменилась по сравнению с топологией

тренажерного комплекса «Белладонна». Тренажерный комплекс обеспечивал одновременную работу четырех тренажеров, что предоставляло принципиальную возможность взаимодействовать четырем тренирующимся экипажам при выполнении совместных комплексных задач. Входящими в состав тренажерного комплекса «Ермак-27» тренажерами можно было управлять с любого из шести *пультов контроля и управления* (ПКУ), так как фактически все приборы-повторители на ПКУ реализованы на форматах в электронном виде. Аппаратная часть вычислительной системы «Ермак-27» спроектирована по модульному принципу, обеспечивающему высокую гибкость при конфигурировании нескольких тренировочных сессий, горячее резервирование вычислительных средств и простоту наращивания комплекса.

Архитектура ВС ТМК *Российского сегмента* (РС) МКС представляет собой дальнейшее развитие идей ВС «Ермак-27». Особенностью комплекса является высокая степень интеграции его систем. Тренажеры имеют общую систему моделирования, в составе комплекса работает общая *модель бортовой вычислительной системы* (МБВС), а управление бортовым оборудованием осуществляется единым терминальным комплексом УСО (рис. 3). Вычислительные средства ТМК РС МКС объединены общими сетями [3].

С 90-х годов XX века в ЦПК стало привлекаться большое количество организаций и предприятий,

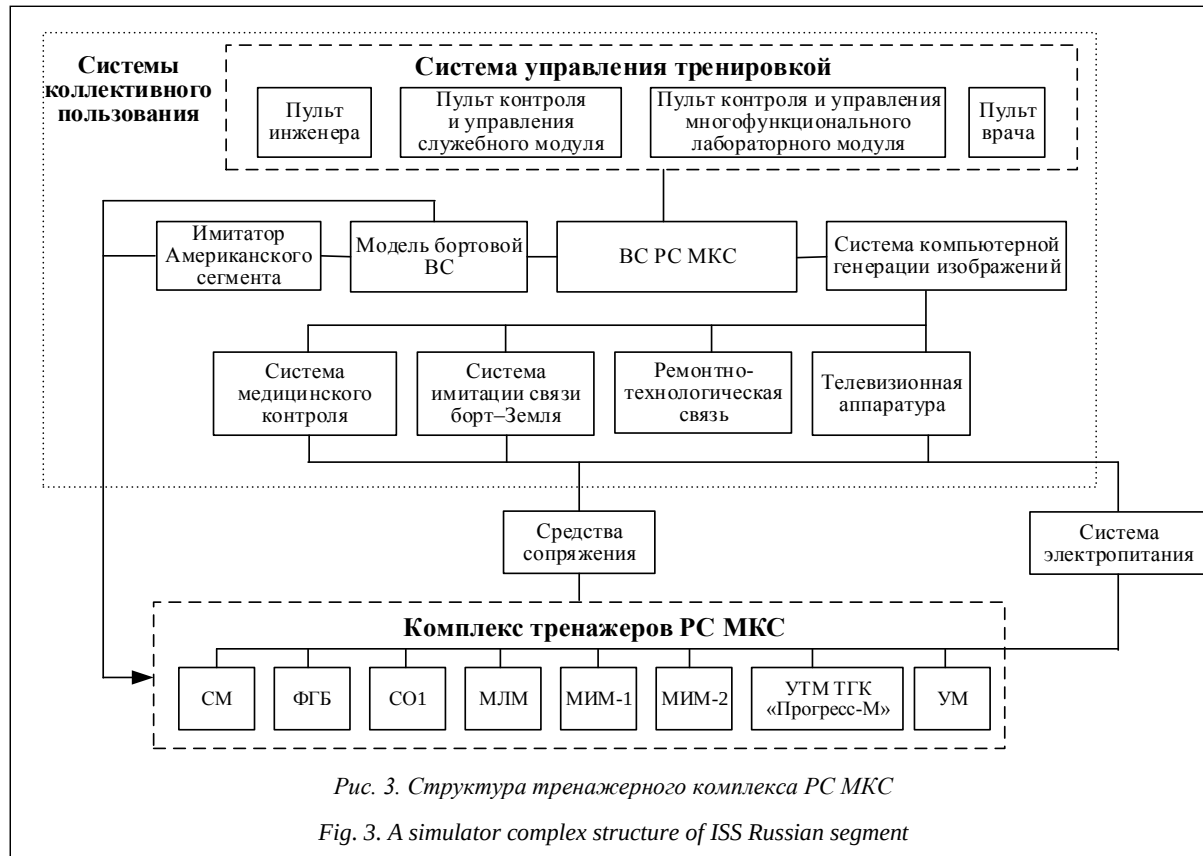


Рис. 3. Структура тренажерного комплекса РС МКС

Fig. 3. A simulator complex structure of ISS Russian segment

создающих ТСПК, что привело к отходу от таких принципов разработки, как унификация, стандартизация, модульность построения, одноразовое проектирование и многократная модернизация космических тренажеров. Следствием этого стало то, что, помимо комплекса РС МКС, начали создаваться локальные (автономные) комплексные и специализированные тренажеры, ФМС.

Таким образом, исторически сложилось так, что комплексы тренажеров космических станций и комплексные тренажеры *транспортных пилотируемых кораблей* (ТПК) создавались обособленно, при этом именно комплексы тренажеров ДОС «Салют», ОК «Мир», РС МКС развивались по идее интеграции программно-технических средств для повышения эффективности их использования.

Тенденция интеграции и объединения ТСПК должна рассматриваться как перспективная и актуальная в настоящее время для ЦПК. Для этого, в первую очередь, нужно обратить внимание на преимущества создания тренажеров в рамках единой системы.

В комплексе тренажеров РС МКС, в который входят в настоящее время восемь тренажеров, используются только три МБВС (разных версий), только два пульта контроля и управления, один пульт врача и ограниченное количество других модулей коллективного пользования. При этом имеется возможность отработки одновременного выполнения в одном или разных модулях нескольких взаимосвязанных полетных операций (а при необходимости и проведения мультиагентных тренировок).

Специализированные тренажеры ТПК «Союз-ТМА» и «Союз-ТМА-2», созданные Центром тренажеростроения и подготовки персонала, также представляют собой единый комплекс, в котором обеспечиваются взаимозаменяемость отдельных систем и одновременная работа экипажей на двух тренажерах, в том числе для отработки выполняемых во взаимодействии задач (например, при возникновении аварийной ситуации и срочном покидании экипажами станции с использованием двух транспортных кораблей), а также при выходе из строя элементов ВС одного из тренажеров. Все это повышает надежность систем, обеспечивая непрерывность подготовки космонавтов, и подтверждает те дополнительные достоинства, которые появляются в ТСПК благодаря интеграционным принципам их построения.

Интеграция – это магистральный путь развития сложных взаимосвязанных систем, в том числе и территориально распределенных систем, ТСПК. Именно поэтому для расширения возможностей создаваемых комплексов, их унификации ставится вопрос о дальнейшей интеграции и формировании в ЦПК распределенной информационно-моделирующей среды, обеспечивающей эффективное и комплексное решение задач теоретической и профес-

сионально-технической подготовки космонавтов на тренажерах и стендах.

Концепция интеграции существующих и эксплуатируемых ТСПК в единый тренажерный комплекс

Решая задачу модернизации ТСПК путем создания единого интегрированного УТМК, следует учитывать то, что практически ничто из ранее созданного в сфере ТСПК, особенно тренажеров, не подлежит замене. В связи с этим методологический подход к модернизации ТСПК в ЦПК должен носить системный характер и осуществляться одновременно по двум направлениям: модернизация ТСПК как способ повышения качества подготовки космонавтов и модернизация как способ повышения эффективности использования ТСПК.

Концептуальной основой модернизации и развития ТСПК является создание распределенной информационно-моделирующей среды, использующей набор универсальных унифицированных интерфейсов и модульный принцип их построения. Необходимо создавать единый интегрированный УТМК, который включал бы в свою структуру как ранее созданные тренажеры и ФМС, так и вновь создаваемые тренажеры и ФМС.

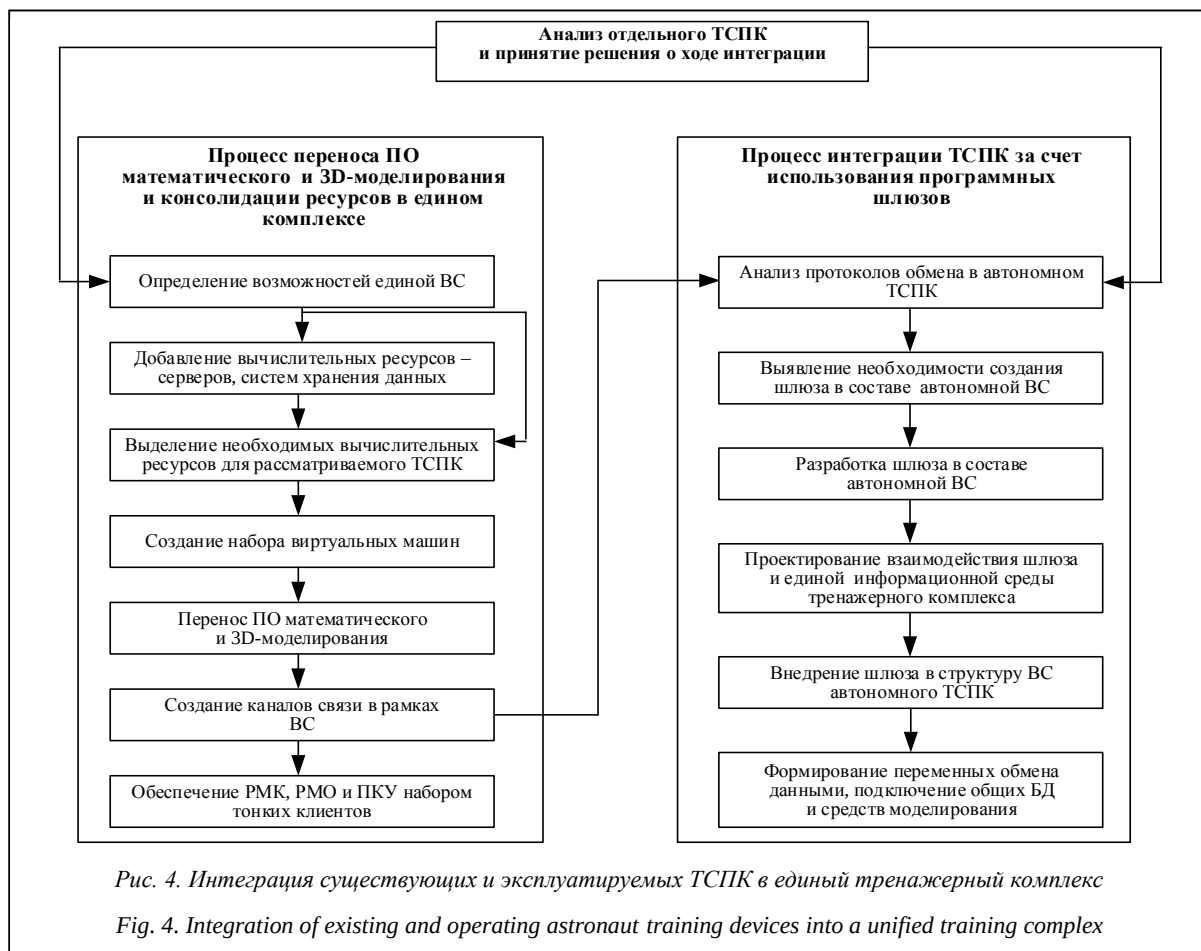
Для этого устанавливаются единые дисциплина и методология в разработке и создании ТСПК, а также исполнение принципов создания новых тренажеров, ФМС и других ТСПК, основанных на модульности, унификации, максимальной стандартизации и коллективном использовании программно-технических средств.

Большое практическое значение имеет разработка подходов к решению проблемы включения в структуру УТМК (или тренажерного комплекса) существующих тренажеров и ФМС, при этом тренировки на них не должны останавливаться. Следует учитывать, что существующее прикладное и специализированное ПО учебно-тренажерной базы ЦПК не должно существенно перерабатываться.

Для решения этих вопросов разработан алгоритм интеграции ТСПК в УТМК (рис. 4), состоящий из двух основных частей: процесса переноса ПО математического и 3D-моделирования и консолидации средств в едином вычислительном комплексе УТМК; процесса интеграции ТСПК за счет использования шлюзов.

Первый вариант в большинстве случаев может быть ориентирован на разработку новых тренажеров, ФМС и средств обучения, а для уже эксплуатируемых в ЦПК ТСПК предложено решать проблему интеграции с помощью шлюзов – программ, обеспечивающих вхождение ранее созданных тренажеров в расширенную информационную среду, единую ЛВС УТМК.

При этом множество не связанных между собой ТСПК объединяются посредством единой системы



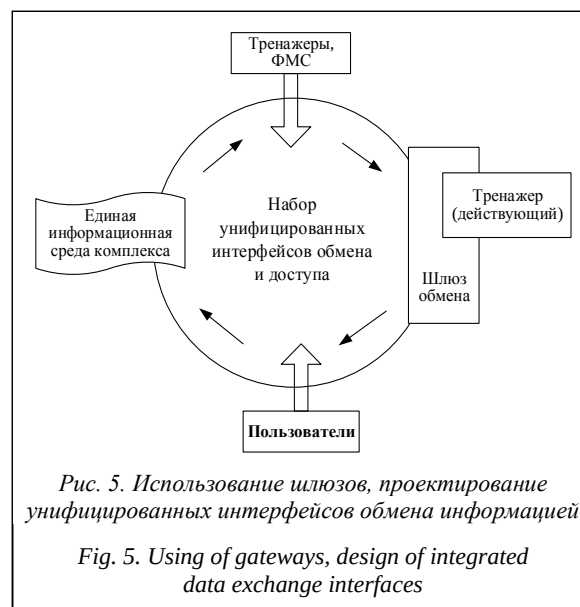
(единая ЛВС, единая БД, единые средства моделирования) с использованием разработанных специализированных программных средств, интегрируемых в качестве шлюза в системы моделирования тренажеров. Множество пользователей получают доступ для работы с выбранными программами, интерфейсами и техническими средствами, входящими в состав учебной и тренажерной баз (рис. 5).

Показанная на рисунке 5 схема отображает использование так называемых шлюзов, которые внедряются в действующие и эксплуатируемые тренажеры.

Шлюз – это специализированное программное средство, обладающее набором унифицированных интерфейсов обмена для обеспечения интеграции информационных потоков тренажера и системы в единую информационную среду.

Реализация каждого конкретного шлюза зависит от набора информационных параметров, которыми будет обмениваться тренажер в процессе своего функционирования с единой информационной средой и другими ТСПК. Таким образом, в ЦПК предлагается концепция трансформации учебных и тренажерных средств подготовки космонавтов в единый интегрированный УТМК посредством использования распределенной сетевой среды моделирования с внедрением ее отдельных компонентов в структуру ВС существующих тренажеров.

Изначально внутренняя интеграция автономных тренажеров достигается применением специализированных ЛВС. После интеграции компоненты единого интегрированного тренажерного комплекса в составе рассматриваемых тренажеров частично используют ВС и сетевую инфраструктуру центрального интегрирующего вычислитель-



ного комплекса. При этом новые тренажеры, полностью построенные с использованием разработанной единой среды моделирования и унифицированных интерфейсов, не требуют включения в свой состав шлюзов, а в существующих тренажерах со своей средой моделирования они необходимы.

Для успешной реализации предлагаемого подхода необходимо использовать новые инновационные технологии, современные технические и программные средства, позволяющие создать единый интегрированный, гибкий, открытый, многофункциональный УТМК нового поколения, который, в свою очередь, обеспечит дальнейшее повышение качества подготовки экипажей космонавтов и эффективность использования всего спектра ТСПК.

Во многом это обусловлено тем, что развитие информационных технологий, а также программной и аппаратной баз ВС вносит изменения в саму концепцию выполнения вычислений, которые становятся распределенными и удаленными от конечного пользователя, в данном случае от обучаемых космонавтов, инструкторов, преподавателей. Все это предполагает применение технологий виртуализации – динамического выделения ресурсов, так называемых облачных вычислений, формирования каналов связи достаточной пропускной способности для передачи конечного результата в виде форматов на экраны пользователей [6].

При решении проблем объединения ТСПК путем создания УТМК нового поколения целесообразно использование единой вычислительной инфраструктуры как базового элемента интеграции множества тренажеров, ФМС, стендов, систем обучения космонавтов с размещением общих БД и моделируемых сущностей в совокупности виртуализированных серверов:

- создание унифицированных рабочих мест ПКУ тренажеров, руководителей обучением в ФМС, рабочих мест обучаемых в ФМС на основе стандартных мебельных конструктивов, с использованием тонких клиентов для отображения форматов и оперативной информации тренажера или ФМС в виде 2D- и 3D-графики, с использованием сенсорных экранов для ввода данных и навигации по управляющим форматам;

- создание единой (общей) ВС, в которой будут размещены все модели бортовых систем, системы визуализации, системы комплексирования, управления тренировками, БД хранения информации об обучаемых и записи проводимых тренировок;

- размещение единой ВС, совместно используемой составными частями УТМК при проведении тренировок и занятий, на аппаратной основе набора вычислительных и графических серверов с подсистемами обеспечения надежности и безопасности.

Именно симбиоз вычислительных и нового типа графических серверов, связанных высокоско-

ростными каналами связи в рамках ЛВС, предоставляет широкие возможности для моделирования, обучения персонала, проведения тренировок, развертывания тренажеров и ФМС различной сложности в рамках единой программно-аппаратной инфраструктуры.

После создания такой единой программно-аппаратной инфраструктуры встает вопрос о технологии использования полученных результатов на рабочих местах, размещенных на территории ЦПК. Для отображения форматов на ПКУ и рабочих местах космонавтов, обучаемых при работе с ФМС, в учебных аудиториях и лабораториях в качестве конечных устройств нужно использовать тонкие клиенты [7]. Это новые устройства без собственных вычислительных мощностей, имеющие возможность отображения и ввода информации с использованием стандартных периферийных устройств – мониторов, клавиатур, джойстиков и т.д. (рис. 6).

Внедрение данных технологий позволит создать единый интегрированный УТМК нового поколения с соответствующей инфраструктурой ЦОД, распределенными технологиями моделиро-



Тонкий клиент Dell Wyse P25
(управление и контроль форматов)



Тонкий клиент Dell Wyse Z90Q8
(3D-графика и высоконагруженное ПО)



Интегрированный тонкий клиент в конструктиве монитора рабочего места обучаемого в ФМС, виртуального тренажера или элемента ПКУ

Рис. 6. Тонкие клиенты

Fig. 6. Thin clients

вания, едиными интерфейсами, средствами обучения и информационной поддержки космонавтов.

В структуру УТМК нового поколения в ЦПК входит ряд взаимосвязанных функциональных комплексов и систем (см. рис. по адресу <http://www.swsys.ru/uploaded/image/2015-4-dop/9.jpg>), решающих определенные задачи, обеспечивающие всестороннюю подготовку космонавтов:

- образовательный комплекс, основанный на применении современных технологий обучения с помощью специализированного ПО для проведения занятий в составе оснащенных мультимедийных аудиторий, учебных классов и лабораторий; комплексы учебно-методических материалов;

- ТМК, включающий как физические, так и виртуальные ТСПК (тренажеры, ФМС, стенды);

- научный комплекс, предназначенный для подготовки космонавтов к проведению научных исследований и экспериментов на борту РС МКС и включающий ряд физических и виртуальных тренажеров и установок;

- система управления тренировками и учебными занятиями, представляющая собой автоматизированную систему планирования и управления процессом подготовки космонавтов и реализующая циклы и алгоритмы проведения типовых учебных сессий;

- специализированные БД и информационные системы, обеспечивающие образовательный, тренировочный, научный и управленческий процессы в составе систем видеонаблюдения, контроля экранов, микротелефонной связи между абонентами, а также системы хранения, образующие единое информационное пространство, на базе которого реализуется всесторонняя подготовка космонавтов;

- система интеграции комплексов в единый УТМК, обеспечивающая не только взаимодействие основных комплексов и систем УТМК, но и формирование вариантов структур и тренажеров из отдельных модулей ТСПК в соответствии с текущими целями и задачами подготовки космонавтов и обеспечения их взаимодействия.

Такой состав комплексов обусловлен в том числе необходимостью интеграции программно-технических средств, используемых, с одной стороны, для образовательного процесса (образовательный комплекс), с другой – для подготовки экипажей космонавтов на тренажерах, тренажерных комплексах, ФМС (ТМК), в единый интегрированный УТМК.

Сейчас становится очевидным, что обучение и профессиональная практическая подготовка как в университетах, так и в центрах подготовки специалистов, например в ЦПК, тесно связаны между собой, они протекают последовательно-параллельно, используют в большинстве своем одну и ту же информацию, одни и те же технические и программные средства; информация, программные и технические средства для подготовки космонавтов

имеют двойное назначение – для общекосмической и профессиональной подготовки.

При использовании средств УТМК на этапе общекосмической подготовки кандидатов в космонавты задействуются средства образовательного комплекса для ознакомления с предметной областью на базе специализированных мультимедийных аудиторий и лабораторий, включающих интерактивные обучающие системы с контролем знаний, возможностями отображения и получения в качестве иллюстрирующего материала при проведении занятий централизованных ресурсов систем наблюдения тренажеров, а также обращения к общим БД учебно-методического материала. На этапе подготовки космонавтов в составе групп используются специализированные и комплексные тренажеры, ФМС из состава программно-технических средств тренажерного и научного комплексов.

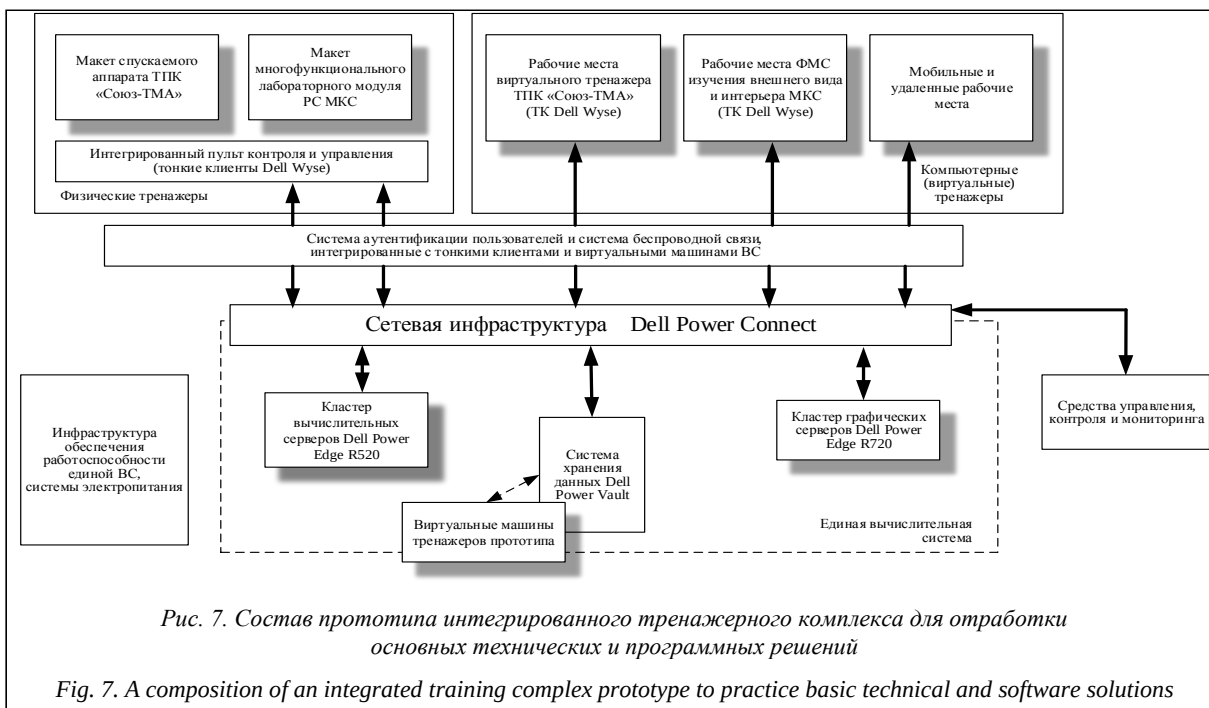
Научный и образовательный комплексы тесно связаны друг с другом посредством общей информационной среды, интегрирующей системы, единых БД, обеспечивающих поддержку учебного процесса. При этом образовательный комплекс обеспечивает, как правило, теоретическую подготовку обучаемых по дисциплинам, научный комплекс обеспечивает проведение экспериментов, демонстрирует работу моделей систем (работа с оборудованием в рамках научных экспериментов на борту орбитальной станции).

Использование специализированного ПО в рамках образовательного комплекса и системы управления тренировками позволит повысить эффективность процессов подготовки и контроля использования компонент учебной и тренажерной баз.

В отличие от классической схемы организации образовательного процесса, построенной на основе разбиения графика обучения на относительно самостоятельные подэтапы (изучение теории, практические занятия, выполнение больших самостоятельных заданий с элементами профессиональной деятельности), при подготовке космонавтов эти этапы ассимилируются: при изучении теории необходимо демонстрировать работу систем, приемы деятельности и так далее. Следовательно, система должна обеспечивать контроль разнообразных взаимосвязанных видов деятельности и возможность корректировки учебных планов в зависимости от результатов обучения. Использование среды управления задачами позволит обеспечить непрерывность контроля уровня подготовленности космонавтов благодаря возможности создания самых разнообразных типов задач для последующей фиксации и оценки прогресса подготовки.

Экспериментальная проверка основных концептуальных решений

Предлагаемые подходы, алгоритмы и решения разработаны и опробованы ООО «Центр тренаже-



ростроения и подготовки персонала» в ходе проведенных НИР и ОКР по данному направлению. Функциональная структура прототипа тренажерного комплекса (рис. 7) отвечает всем требованиям разработанной концепции. В состав прототипа входят аналог комплексного тренажера – тренажер ТПК «Союз-ТМА» на основе полномасштабного макета спускаемого аппарата, аналог тренажеров комплекса РС МКС – многофункциональный лабораторный модуль, оснащенный телеоператорным режимом, аналог специализированного тренажера – виртуальный (компьютерный) тренажер сближения и стыковки ТПК «Союз-ТМА», аналог ФМС на основе средств виртуальной реальности – учебный класс изучения внешнего вида и интерьера МКС.

На данном прототипе проверялись прежде всего параметры, характеризующие производительность, адекватность и качество отработки всего специализированного ПО, скорости обмена, загрузки оборудования и сетевых каналов.

Следует отметить, что до интеграции в единый комплекс каждая из составных частей прототипа имела собственную автономную вычислительную систему, построенную на стандартных персональных компьютерах, тренажеры имели автономные собственные ПКУ или рабочие места инструкторов, также построенные на персональных компьютерах различных типов.

При интеграции и создании прототипа проверялись принципы построения на основе концептуальной модели, включающей распределенную информационно-моделирующую среду; проверялся принцип переноса специализированного ПО математического и 3D-моделирования в структуру интегрированного вычислительного комплекса; принципы замены ПК в тренажерах, ПКУ на тон-

кие клиенты. Для этого использовались самые современные инновационные технологии передовых фирм, таких как Dell, VMware и Citrix – это применение высокопроизводительных серверов, технологии выделения графических ресурсов серверов NVIDIA GRID, специализированное ПО систем виртуализации.

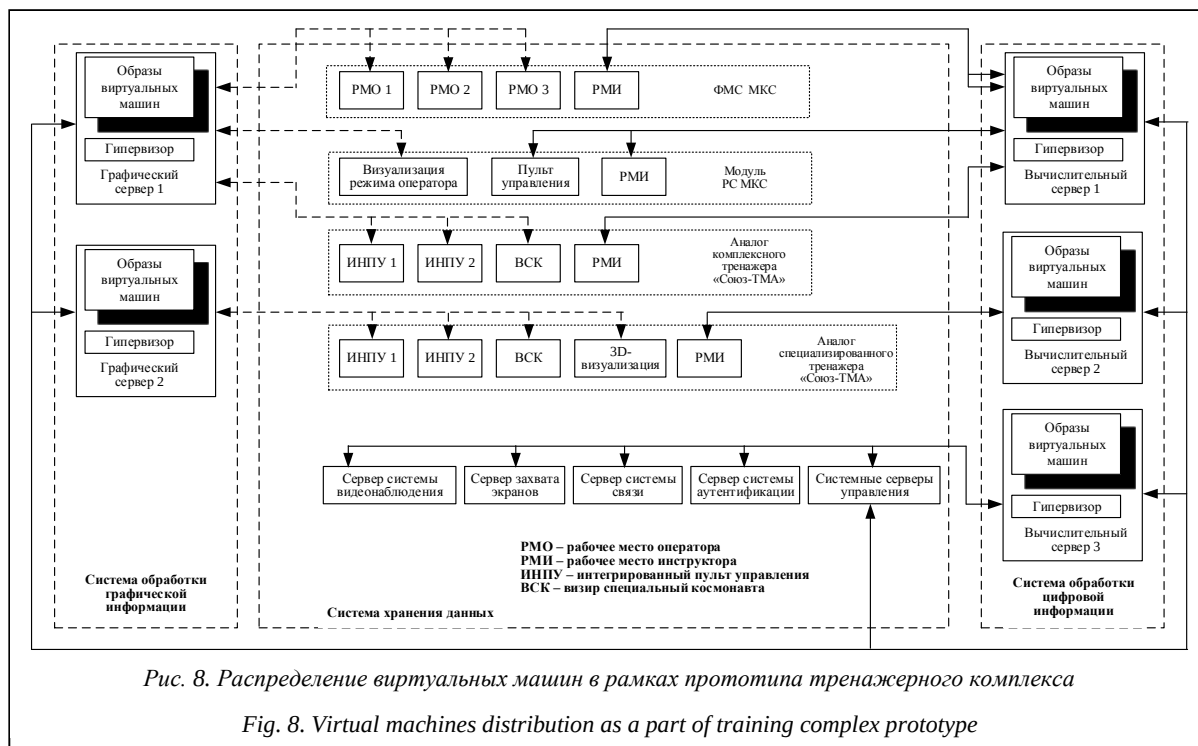
Исходя из требований к производительности ПО тренажеров были созданы соответствующие виртуальные машины (рис. 8) – программные компьютеры, у которых есть свои процессоры, память, дисковое пространство. Как видно на рисунке, виртуальные машины распределяются между вычислительными и графическими серверами.

Основные результаты экспериментальной проверки прототипа:

- структурно созданный прототип включает распределенную информационно-моделирующую среду и единый вычислительный комплекс, рабочие места космонавтов в составе аналогов комплексных и специализированных тренажеров и ФМС, интегрированного ПКУ управления – здесь подтверждается принцип построения единого комплекса на основе сформированной концептуальной модели;

- функционирование ПО математического и 3D-моделирования в едином вычислительном комплексе на созданных конфигурациях виртуальных машин вычислительных и графических серверов подтверждает принцип переноса специализированного ПО в структуру интегрированного вычислительного комплекса и включения в единый комплекс ранее созданных ТСПК;

- использование в качестве устройств доступа к серверным ресурсам на РМК, ПКУ и РМО в стендах тонких клиентов подтверждает принцип за-



мены ПК в тренажерах и ФМС на данные устройства без снижения производительности и качества получаемой на средствах отображения визуальной информации;

- с установленными параметрами виртуальных машин вычислительные ресурсы сервера и оперативная память используются на 70 %, использование же процессорного времени зависит от интенсивности использования тренажеров, но не превышает 50 %; максимальная сетевая загрузка (порядка 150 Кбит/сек.) достигнута при вызове форматов управления и передаче команд – тест на вычислительном сервере, что доказывает имеющийся запас выделенных ресурсов и достаточную низкую загрузку сетевых каналов, которая не может влиять на недостатки производительности и задержки передачи данных по сети;

- в рабочем состоянии графического сервера рассматриваемых тренажеров и ФМС средняя нагрузка на центральном процессоре колеблется от 30 до 50 %, мгновенные значения могут достигать 70 %; при получении данных с графического сервера наиболее активный сетевой обмен происходит при подключении к виртуальной машине (момент загрузки тонкого клиента и инициализации подключения), то есть именно тогда, когда передается начальный трафик всей рабочей области, отображаемой на мониторе, далее идет оптимизированный поток данных по изменениям отдельных элементов; средняя скорость обмена составляет от 1 до 5 Мбит/сек., но мгновенные значения могут достигать 20 Мбит/сек. (при резких изменениях отображаемого изображения на экране, в системе моделирования или действиях оператора) – все это также

доказывают имеющийся запас выделенных ресурсов и показатели загрузки сетевых каналов, позволяющие говорить о достаточной скорости и качестве каналов связи при наличии ЛВС не хуже Gigabit Ethernet в рамках тренажерного комплекса;

- показанный прототип интегрированной ВС тренажерного комплекса, состоящий из двух кластеров (3 вычислительных и 2 графических сервера), включает в среднем по 10 виртуальных машин тренажерного комплекса, при этом общая загрузка серверов составила не более 40–60 %.

Предлагаемая концепция вполне реализуема, и это заявление базируется на проведенных экспериментах, подтвердивших реализуемость ее основных принципиальных положений. В концепции разработки единого интегрированного УТМК заложена возможность его развития без изменения принципиальных основ, без радикального перепроектирования при включении в его состав новых тренажеров, ФМС, учебно-методических средств с минимальными затратами ресурсов.

УТМК, принципы его создания, применяемые современные информационные технологии, программные и технические средства обеспечивают переход в создании тренажеров, ФМС и других технических средств подготовки и обучения космонавтов, входящих в его состав, к глубокой унификации, модульности построения программно-технических средств, что гарантирует сокращение времени на разработку, проектирование, создание и модернизацию новых тренажеров, тренажерных комплексов и технических средств обучения.

В УТМК обеспечивается возможность использования ЦОД, которые повышают эффективность

использования вычислительных и графических ресурсов, надежность функционирования УТМК в целом и входящих в него специализированных и комплексных тренажеров и других ТСПК.

Основные концептуальные решения неоднократно обсуждались на конференциях и научно-технических советах, проводимых в НИИ ЦПК, а технические решения на основе применения передового оборудования фирмы Dell – на семинарах российского представительства компании Dell. Высокую оценку пилотное внедрение получило также от ведущих мировых компаний в области виртуализации рабочих столов и использования технологии VDI – компании Citrix и лидера в области визуальных вычислений и производства графических карт – компании NVIDIA [8–10].

Литература

1. Шукшун В.Е., Янышкин В.В. Учебно-тренажерно-моделирующие комплексы нового поколения для подготовки космонавтов. М.: Машиностроение, 2015. 112 с.

2. Центр подготовки космонавтов имени Ю.А. Гагарина. Официальный сайт. URL: <http://www.gctc.ru> (дата обращения: 17.08.2015).

3. Шукшун В.Е., Циблиев В.В., Потоцкий С.И. Тренажерные комплексы и тренажеры. Технологии разработки и опыт эксплуатации. М.: Машиностроение, 2005. 384 с.

4. Наумов Б.А., Шевченко Л.Е. Космические тренажеры. Этапы развития. Звездный городок: Изд-во НИИ ЦПК им. Ю.А. Гагарина, 2008. 178 с.

5. Наумов Б.А. Космические тренажеры. Звездный городок: Изд-во НИИ ЦПК им. Ю.А. Гагарина, 2013. 214 с.

6. Анализ современных технологий виртуализации. URL: <http://habrahabr.ru/company/centosadmin/blog/212985/> (дата обращения: 17.08.2015).

7. Продукция. URL: http://www.wiat.ru/products/tonkie_klienty/wyse/ (дата обращения: 17.08.2015).

8. Citrix и NVIDIA | NVIDIA. URL: <http://www.nvidia.ru/object/citrix-desktop-vdi-case-studies-ru.html> (дата обращения: 17.08.2015).

9. Citrix и NVIDIA провели семинар для сервис-провайдеров. URL: <http://www.bytemag.ru/press/detail.php?ID=27853> (дата обращения: 17.08.2015).

10. IT Solutions – Center of Simulator and Training – Case Study – Citrix. URL: <http://www.citrix.ru/customers/center-of-simulator-and-training-ru.html> (дата обращения: 17.08.2015).

DOI: 10.15827/0236-235X.112.005-015

Received 09.09.15

CONCEPTUAL FRAMEWORK FOR DEVELOPMENT OF THE NEXT-GENERATION EDUCATIONAL TRAINING AND SIMULATION COMPLEX

Shukshunov V.E., Dr.Sc. (Engineering), Professor, Director General, v.shukshunov@simct.ru;

Yanyushkin V.V., Ph.D. (Engineering), Deputy General Manager, v.yanyushkin@simct.ru

(The Center for Space Simulator Development and Personnel Training,

Pervomayskaya St. 92, Moscow, 115088, Russian Federation)

Abstract. The article considers various approaches to the development of a new-generation educational training and simulation complex (ETSC) at Gagarin Research & Test Cosmonaut Training Center using the latest technical and software tools. The article offers the approaches to decomposition of the ETSC development tasks by several interrelated stages solving certain problems. The article provides a short analysis of historical facts covering several generations of astronaut training complexes including integrating principles implementation. The paper describes the algorithm based on two major approaches to integrating existing tech astronaut training devices into one simulator complex. The authors consider a concept of gateway and modern computing technologies along with virtualization technologies and a set of hardware & software means. The article provides the description of the promising ETSC complex structure; each complex is a separate part that solves a certain number of problems. Special attention is given to the integration of educational and training processes while preparing specialists. Finally, the paper provides the composition of the experimental stand for testing all offered solutions connected to the ETSC development. There are the descriptions of obtained results that reflect absolutely new solutions including the use of host-centric computations of a unified computing system and the use of thin clients at the workplaces. The authors analyze the level of resources loading, data transfer network channels and other critical parameters. It is noted that many IT experts highly evaluated the results of the work for development of experimental stand.

Keywords: educational training and simulation complex, resource virtualization technology, astronaut training devices, autonomous simulators integration, unified software & hardware infrastructure.

References

1. Shukshunov V.E., Yanyushkin V.V. *Uchebno-trenazhernye modeliruyushchie komplekсы novogo pokoleniya dlya podgotovki kosmonavtov* [New Generation Modeling and Simulating Systems for Training Astronauts]. Moscow, Mashinostroenie Publ., 2015, 112 p.

2. *Centr podgotovki kosmonavtov imeni Yu.A. Gagarina* [Gagarin Research&Test Cosmonaut Training Center. Official Web-site]. Available at: <http://www.gctc.ru> (accessed August 17, 2015).

3. Shukshunov V.E., Tsibliyev V.V., Pototsky S.I. *Trenazhernye komplekсы i trenazhery. Tekhnologii razrabotki i opyt ekspluatatsii* [Training Complexes and Simulators. Design Technique and Field Experience]. Moscow, Mashinostroenie Publ., 2005, 384 p.

4. Naumov B.A., Shevchenko L.E. *Kosmicheskie trenazhery. Etapy razvitiya* [Space Simulators. Development Stages]. Star City, 2008, 178 p.

5. Naumov B.A. *Kosmicheskie trenazhery* [Space Simulators]. Star City, 2013, 214 p.

6. *Analiz sovremennykh tekhnology virtualizatsii* [Analysis of Modern Virtualization Technologies]. Available at: <http://habrahabr.ru/company/centosadmin/blog/212985/> (accessed August 17, 2015).

7. *Produktsiya* [Production]. Available at: http://www.wiat.ru/products/tonkie_klienty/wyse/ (accessed August 17, 2015).

8. *Citrix i NVIDIA | NVIDIA* [Citrix and NVIDIA]. Available at: <http://www.nvidia.ru/object/citrix-desktop-vdi-case-studies-ru.html> (accessed August 17, 2015).

9. *Citrix i NVIDIA proveli seminar dlya servis-provayderov* [Citrix and NVIDIA Held a Workshop for Service-Providers]. Available at: <http://www.bytemag.ru/press/detail.php?ID=27853> (accessed August 17, 2015).

10. *IT Solutions – Center of Simulator and Training – Case Study – Citrix*. Available at: <http://www.citrix.ru/customers/center-of-simulator-and-training-ru.html> (accessed August 17, 2015).

УДК 004.72

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.016-021

ОСНОВЫ АНАЛИЗА ИТ-ИНФРАСТРУКТУРЫ ОТКРЫТОГО КОНТУРА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ИНФОРМАЦИОННОГО ОБЕСПЕЧЕНИЯ ПОДГОТОВКИ КОСМОНАВТОВ ДЛЯ СОЗДАНИЯ ИНТЕГРИРОВАННОГО ТРЕНАЖЕРНОГО КОМПЛЕКСА

*С.Н. Ковригин, начальник отдела, S.Kovrigin@gctc.ru
(НИИ ЦПК им. Ю.А. Гагарина, Звездный городок, Московская обл., 141160, Россия);
В.В. Янюшкин, к.т.н., зам. генерального директора, v.yanyushkin@simct.ru
(Центр тренажеростроения и подготовки персонала,
ул. Первомайская, 92, г. Москва, 115088, Россия)*

В работе описан предмет тестирования открытого контура автоматизированной системы информационного обеспечения подготовки космонавтов НИИ ЦПК имени Ю.А. Гагарина с учетом создания тестового стенда платформ виртуализации VMware и Citrix с дальнейшим размещением набора тонких клиентов в рамках разветвленной сетевой структуры. Кратко описываются структура стенда и используемые технические решения, в том числе графический сервер, оснащенный платами NVIDIA GRID K2, тонкие клиенты системы виртуализации двух типов и ПО системы виртуализации. В качестве объектов тестирования выбирается специализированное ПО двух компонент технических средств подготовки космонавтов: рабочие места виртуальной Международной космической станции и рабочие места обучаемых автономного тренажера «Телеоператор», оснащенного ручками управления движением и навигацией. Работа данных средств проверяется в рамках систем VMware и Citrix. Эксперименты должны подтвердить возможность развертывания систем виртуализации тренажеров и функционально-моделирующих стендов в рамках описываемой автоматизированной системы. В работе приводятся основные технические и концептуальные решения, подвергающиеся проверке, а также предложения для разработки программно-аппаратной инфраструктуры закрытого контура на основе современных технологий. Четко определены основные требования, которым должны отвечать массивы серверов, системы хранения, коммутационное оборудование и конечные устройства. Особенностью работы является то, что все тестирование проводится на базе потенциального заказчика, на его сетевой инфраструктуре и существующих рабочих местах. Целью выполнения работы является определение возможности создания набора интегрированных решений для консолидации ПО тренажеров функционально-моделирующих стендов в рамках корпоративного центра обработки данных.

Ключевые слова: автоматизированная система информационного обеспечения подготовки космонавтов, интегрированный комплекс, центр обработки данных, сетевая инфраструктура, графический сервер.

Разветвленная инфраструктура автоматизированной системы информационного обеспечения подготовки космонавтов (АСИОПК) создана в НИИ ЦПК имени Ю.А. Гагарина в 2009–2014 гг. [1]. С целью оценки возможности использования ИТ-инфраструктуры АСИОПК для реализации единого интегрированного учебно-тренажерно-моделирующего комплекса (УТМК) предполагается проведение работ по тестированию и отработке ряда новых программных и технических решений.

ИТ-инфраструктура АСИОПК основывается на трехуровневой системе передачи данных и едином виртуальном вычислительном комплексе (ВВК), построенном с использованием системы виртуализации компаний VMware, изделий и ПО компаний Veeam, HP и NetApp. Система передачи данных общего информационного обмена (открытого сегмента) АСИОПК (рис. 1) состоит из уровней доступа (access level) и распределения (distribution level), а также ядра (core level) [2].

Так, уровень доступа предназначен для проводного подключения конечных пользовательских устройств, имеющих сетевые интерфейсы (компьютеры, принтеры, wi-fi-точки доступа и пр.) к остальным ресурсам сети. Уровень доступа состоит из одного или более (в стеке) коммутаторов и структурированной кабельной системы. В каче-

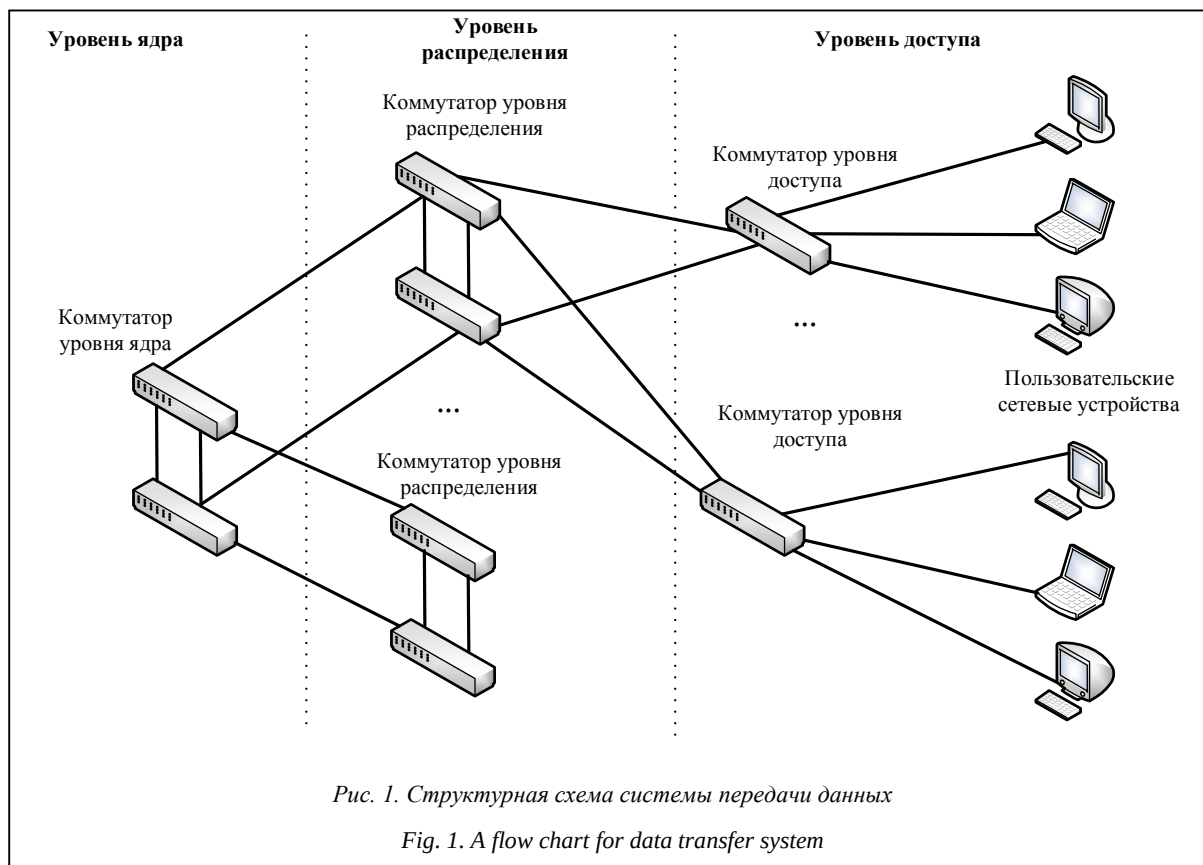
стве сетевых коммутаторов используются модели типа Huawei S3352P-SI. Для обеспечения отказоустойчивости каждый коммутатор уровня доступа подключен двумя (медными и/или оптическими) кабелями к коммутатору уровня распределения.

Уровень распределения предназначен для объединения коммутаторов уровня доступа в группы и подключения их к магистральной линии связи. Уровень распределения состоит из пары (стека) коммутаторов Huawei S5328C-EI-24S. Каждый коммутатор уровня распределения подключен оптическим кабелем к коммутатору уровня ядра.

Уровень ядра предназначен для объединения всех подсетей и маршрутизации между ними. В качестве маршрутизирующих коммутаторов используются два территориально разнесенных между собой коммутатора Huawei S9306.

Вычислительным ядром АСИОПК является виртуальный вычислительный комплекс, построенный на базе четырех серверов HP ProLiant DL380 G6, системы хранения NetApp, ПО виртуализации VMware [3] и резервного копирования Veeam [4]. На все серверы установлены гипервизоры VMware ESXi 5.1.0, объединенные в единую виртуальную инфраструктуру с помощью ПО VMware vCenter.

ПО функционально-моделирующих стендов (ФМС), комплексных и специализированных тре-



нажеров, то есть *технических средств подготовки космонавтов* (ТСПК), дополнительно к требованиям безопасности накладывает жесткие ограничения на вычислительный комплекс и всю ИТ-инфраструктуру относительно времени передачи информации.

С целью проведения оценки возможности использования информационной инфраструктуры АСИОПК для реализации единого интегрированного УТМК предлагается провести работы по оценке характеристик информационной инфраструктуры АСИОПК, используя систему передачи данных открытого контура. Для этого должен быть создан тестовый стенд и проведен ряд экспериментов, подтверждающих корректную работу специализированного ПО тренажеров и ФМС в рамках систем виртуализации.

Задачи и структура тестового стенда

В качестве сервера системы виртуализации используется графический сервер Dell PowerEdge 720, оснащенный картами NVIDIA GRID K2.

Новейшая серверная технология, которая использовалась компанией Dell при разработке этого сервера, обеспечивает отличные возможности внедрения виртуализации, обработки больших объемов информации. Сервер Dell PowerEdge R720 обладает прекрасной плотностью записи, обеспечивает сбалансированную производительность

ввода-вывода, а также вычислительную мощность современных процессоров [5].

Производительные процессоры в сочетании с достаточным объемом оперативной памяти обеспечивают высокую эффективность серверных приложений. С помощью сервера Dell PowerEdge R720 можно увеличить количество виртуальных машин на одном сервере и существенно расширить возможности виртуализированной среды. Возможность установки дополнительного GPU-ускорителя, который оказывает дополнительную вычислительную поддержку при высокопроизводительных вычислениях, также помогает увеличить производительность технологии VDI [6].

В качестве ПО для виртуализации используется VMware VSphere 6 в редакции Enterprise Plus и Citrix XenServer [7], данные версии ПО поддерживают все последние технологии, в том числе NVIDIA.

В роли специализированного ПО для проверки ИТ-инфраструктуры АСИОПК выступают виртуальные машины стенда системы виртуальной реальности, включающие в себя три рабочих места обучаемых пользователей и одно рабочее место преподавателя, и виртуальные машины автономного тренажера, включающие в себя визуализацию режима, виртуальный пульт управления и рабочее место инструктора (рис. 2).

В качестве технологии виртуализации GPU на обеих платформах виртуализации используется

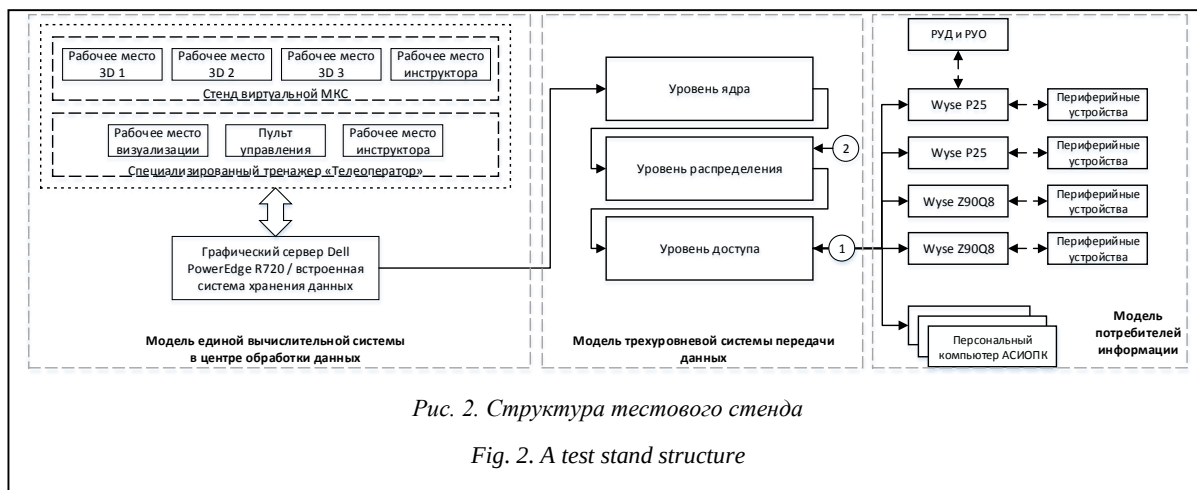


Рис. 2. Структура тестового стенда

Fig. 2. A test stand structure

технология NVIDIA GRID vGPU. NVIDIA создала технологию NVIDIA GRID vGPU, обеспечивающую прямую передачу графических команд каждой виртуальной машины в GPU, без трансляции гипервизором. Один физический графический модуль, благодаря драйверу, имеет несколько профилей виртуального GPU с различным количеством ядер и дискретной графической памяти. Именно данное преимущество, в отличие от других существующих технологий виртуализации графических ресурсов, даст возможность более оптимального распределения и использования дорогостоящих ресурсов графических серверов.

Для проведения тестирования используются специализированные тонкие клиенты Wyse P25 и Wyse Z90Q8 [8].

Wyse P25 с аппаратной поддержкой PCoIP (PC-over-IP, personal computer over internet protocol – персональный компьютер по протоколу IP) [9] для VMware View представляет собой нулевой клиент, обеспечивающий графическую производительность для таких сложных приложений, как CAD (проектирование) и 3D-моделирование в реальном времени. Клиент разработан специально для VMware View, с процессором от Teradici и аппаратной поддержкой протокола PCoIP, включая кэширование на стороне клиента для достижения высокого уровня производительности. Подключается клиент с использованием сетевого интерфейса Ethernet, имеет 4 порта USB для подключения и проброса устройств, поддержку до 2 мониторов, при этом потребляя менее 8 Вт электроэнергии.

В отличие от программных реализаций выделенный аппаратный механизм PCoIP обеспечивает высокий уровень производительности графической системы для приложений, таким образом, клиент Wyse P25 обладает всеми преимуществами эффективной защищенной централизованной вычислительной среды единого центра обработки данных (ЦОД). Оптимизированные алгоритмы сжатия работают в режиме реального времени, обеспечивая максимальное качество изображений для доступной пропускной способности сети.

Клиент Wyse Z90Q8 – это высокопроизводительный тонкий клиент, оснащенный средствами ускорения обработки графики и мультимедиа [10].

Дополнительная вычислительная мощность обеспечивает поддержку виртуального рабочего стола Windows и облачных приложений с широкими возможностями создания и обработки содержимого, воспроизведения видео высокого разрешения и работы с 3D-графикой. Клиенты данной серии оснащены операционной системой Windows® Embedded Standard.

При проведении тестирования посредством работы автономного тренажера тонкий клиент дополнительно оснащается действующими макетами ручек (управления движением и управления ориентацией), подключенными по интерфейсу USB. Это обеспечивает отработку подключения внешних устройств к тонкому клиенту и оценку времени реакции на управление объектами с помощью данных манипуляторов.

Также предполагается использование при проведении проверок программных тонких клиентов, установленных на персональные компьютеры существующих рабочих мест, пользователей АСИОПК.

Состав работ и проводимые эксперименты

На тестовый стенд устанавливается и настраивается система виртуализации, разворачиваются необходимые виртуальные серверы и проводится тестирование с использованием тонких клиентов, выполняя подключение к системе передачи данных АСИОПК на различных уровнях (рис. 2 – отметки 1 и 2).

Таким образом оценивается возможность использования существующих каналов локальной вычислительной сети (ЛВС), волоконно-оптических линий связи и средств доступа АСИОПК в интересах реализации интегрированных ТСПК. Параллельно оцениваются возможности комплекса технических и программных средств ИТ-инфраструктуры АСИОПК при использовании систем

виртуализации и доступа к результатам вычислений виртуальными машинами ТСПК из ЦОД.

Для более детального анализа сравниваются результаты работы специализированного ПО и возможности систем виртуализации Citrix и VMware, а также осуществляется проверка возможности переноса ПО вычислительных систем ФМС и автономного тренажера на ИТ-инфраструктуру АСИОПК с сохранением существующего функционала.

Во время работы комплекса виртуальных машин каждого из тренажеров средствами мониторинга системы виртуализации фиксируется нагрузка на выделенные для данных виртуальных машин ресурсы, средствами системы мониторинга System Center Operations Manager проводится оценка загрузки портов коммутаторов, к которым подключены используемый тонкий клиент и аппаратная часть сервера виртуализации.

Во время работы пользователей посредством тонких клиентов с комплексом виртуальных машин каждого из тренажеров должна проводиться оценка задержки передачи данных от клиента к виртуальной машине и от виртуальной машины к тонкому клиенту, а также качества изображения, количества кадров в секунду и времени задержки при использовании устройств ввода.

Требования к программной и аппаратной составляющим закрытого контура

При создании инфраструктуры закрытого контура в части интегрированного тренажерного комплекса должно быть предусмотрено решение следующих задач:

- создание и выделение виртуализированных вычислительных серверов из общего набора ресурсов с заданными операционными системами, оперативной памятью и процессорным временем для размещения на них систем моделирования, СУБД, интегрирующего ПО тренажеров и ФМС;
- создание и выделение виртуализированных графических серверов из общего набора ресурсов с заданными операционными системами, оперативной и графической памятью и процессорным временем для размещения на них систем 3D-моделирования, выполнения сложных графических расчетов в реальном времени;
- отработка взаимодействия наборов виртуальных серверов с удаленными клиентами посредством проводных и беспроводных ЛВС АСИОПК, а также набора персональных компьютеров и мобильных клиентов для получения и отображения форматов управления, данных 3D-моделирования и другой вспомогательной информации;
- отработка систем управления структурой созданных вычислительных и графических сетевых сервисов, в том числе резервирования, обеспечения надежности, антивирусной защиты, опера-

тивного контроля и мониторинга производимых локальными и удаленными клиентами операций.

В соответствии с перечисленными задачами к составным частям ЦОД закрытого контура предъявляются следующие основные требования:

- 1) предоставление возможности эффективного взаимодействия проектируемых сетевых сервисов для организации работы специализированного ПО тренажеров, систем виртуальной реальности и вспомогательных систем;
 - 2) предоставление возможности гибкого масштабирования информационной системы при наращивании ресурсов и изменении конфигурации системы;
 - 3) обеспечение отказоустойчивости на уровне выхода из строя любого сервера, коммутационного узла, канала связи;
 - 4) обеспечение отказоустойчивости на уровне выхода из строя одного накопителя данных в любом узле системы;
 - 5) наличие надежной системы резервного копирования информационных БД и виртуальных серверов;
 - 6) обеспечение оборудования ЦОД электричеством в течение 15–20 минут после отключения общего электроснабжения;
 - 7) возможность установки различных уровней доступа для индивидуальных пользователей или групп пользователей при работе с приложениями и ресурсами;
 - 8) разбивка стендов и развертываемых тренажеров на соответствующие сетевые сегменты, возможно, виртуальные частные сети;
 - 9) в серверном сегменте, кроме самих серверов, допускается нахождение только компьютеров администрирования и контроля, если таковые будут предусмотрены проектом закрытого контура;
 - 10) обеспечение защиты локальной сети закрытого контура АСИОПК (а также виртуальных машин и сетей) от внешних сетевых угроз Интернета, а также от внутренних угроз открытого контура и организационными, и программными мерами.
- Для реализации требований 1 и 2 система должна быть оснащена набором высокопроизводительных серверов, поддерживающих технологии виртуализации, а также соответствующим сетевым оборудованием.
- Для реализации требований 3–5 система должна иметь возможность построения RAID-массивов для резервного хранения информации, а также оснащаться общим сетевым хранилищем. Дополнительным решением может быть введение дублирующих элементов сетевого оборудования и соответствующих программных средств.
- Для реализации требования 6 система должна быть оснащена источниками бесперебойного питания.
- Для реализации требований 7 и 8 система поддерживает виртуализацию ресурсов, а также созда-

ние виртуальных частных сетей на уровне серверов и активного сетевого оборудования.

Для реализации требования 9 система должна быть оснащена рабочим местом администрирования, контроля комплекса, а также необходимым ПО, управляемым с данного рабочего места.

Для реализации требования 10 система должна включать антивирусную защиту, средства ограничения доступа в глобальные сети и внутренние процедуры восстановления работоспособности, а также административные меры контроля доступа в помещения.

Для повышения надежности и единообразия используемых аппаратных средств предпочтение должно быть отдано уже апробируемым и используемым в открытом контуре продуктам ведущих мировых компаний.

Прикладное ПО закрытого контура предназначено для осуществления следующих операций: управление виртуальной инфраструктурой серверов; создание резервных копий в случае возникновения критических ошибок и обеспечение возможности отката; контроль энергопитания и автоматического выключения оборудования в случае возникновения аварийных ситуаций; обеспечение антивирусной защиты и всего спектра внутренних и внешних угроз для ПО; оперативный мониторинг и отслеживание сетевой активности пользователей с целью контроля и предотвращения несанкционированных действий; обеспечение работоспособности сетевых служб, таких как Active Directory, DNS, DHCP; настройка и администрирование набора пользовательских сервисов и специализированного ПО.

Прикладное ПО серверов обработки информации представляет собой набор специализированных приложений системы управления тренировками, интеграции ПО, моделирования и визуализации объектов тренажеров и ФМС, а также для хранения БД и вспомогательной информации.

Требования к серверам. Высокопроизводительный вычислительный сервер предназначен для развертывания виртуальных серверов с задачами и сервисами, выдачи вычислительных мощностей для виртуальных машин обрабатываемых стендов и тренажеров. Данный сервер должен обладать внутренним аппаратным RAID-контроллером, сетевым контроллером с несколькими портами по 1 Гбит/с и возможностью установки модулей до 10 Гбит/с, системой хранения данных, возможностью установки достаточных объемов оперативной памяти. Высокопроизводительный графический сервер предназначен для развертывания виртуальных серверов с задачами и сервисами детализированной 3D-графики, выдачи вычислительных мощностей для виртуальных машин обрабатываемых стендов и тренажеров. Сервер должен обладать достаточным объемом оперативной памяти и иметь в своем составе графические карты, позволяющие

выделять ресурсы для создания виртуальных графических адаптеров со всеми возможностями физических адаптеров. Система сетевого хранения должна обеспечивать хранение информации, образов виртуальных машин, подключаемых дисковых массивов к системам, БД с возможностями резервирования и создания любых конфигураций RAID-массивов. Для сетевого взаимодействия должны использоваться высокопроизводительные коммутаторы серверного сегмента, которые обеспечивают объединение компонент серверного сегмента, системы хранения и серверов и поддерживают большую скорость внутренней передачи.

Исходя из требований к производительности ПО тренажеров должны быть созданы соответствующие виртуальные машины. Все образы виртуальных машин должны храниться в единой системе хранения данных для обеспечения централизованного доступа и удобства перемещения ВМ между серверами.

В качестве платформы виртуализации может выступать VMware vSphere или Citrix XenServer, которые на данный момент поддерживают технологию GRID Virtual GPU, но, так как в открытом контуре применяется ПО VMware vSphere, для сохранения унификации серверных кластеров и при положительных оценках системы виртуализации на специализированном ПО тренажеров и ФМС предпочтительнее выбор системы VMware vSphere.

При развертывании системы виртуализации серверных мощностей в VMware vCenter должны быть созданы кластеры и включены службы HA и DRS, обеспечивающие миграцию работающих виртуальных машин в случае отказа и для распределения равномерной нагрузки на серверы.

Специализированный протокол обмена системы виртуализации PCoIP определяет низкие требования к пропускной способности сетевых каналов обмена между набором тонких клиентов на тренажерах и ФМС и единым вычислительным комплексом УТМК. При этом используется компрессированный поток реального времени для видеоданных и трафика USB устройств в виде IP-пакетов для передачи на конечное устройство отображения информации только об изменившихся пикселях изображения.

Для этого должны применяться тонкие клиенты на рабочих местах АСИОПК закрытого контура. В части выдвигаемых требований к тонким клиентам на рынке представлены модификации нулевых клиентов для основных платформ виртуализации (Citrix и VMware), которые обладают всеми необходимыми характеристиками и достаточным уровнем производительности в зависимости от решаемой задачи. Следует отметить, что при работе с технологией VDI и результатами удаленных графических расчетов, а также при использовании системы виртуализации VMware желательнее исполь-

зование клиентов, поддерживающих на аппаратном уровне протокол PCoIP, например Wyse P25 или P45.

Литература

1. Центр подготовки космонавтов имени Ю.А. Гагарина. Официальный сайт. URL: <http://www.gctc.ru> (дата обращения: 18.08.2015).
2. Основы коммутации. URL: <http://www.intuit.ru/studies/courses/3591/833/lecture/14251?page=9> (дата обращения: 18.08.2015).
3. Виртуализация VMware для настольных компьютеров, серверов, приложений, общедоступных и гибридных облаков. URL: <http://www.vmware.com/ru> (дата обращения: 18.08.2015).
4. Резервное копирование VMware Sphere. URL: <http://www.veeam.com/ru/vmware-esx-backup.html> (дата обращения: 18.08.2015).

www.veeam.com/ru/vmware-esx-backup.html (дата обращения: 18.08.2015).

5. Подробные сведения о сервере для установки в стойку PowerEdge R720. URL: <http://www.dell.com/ru/business/p/power-edge-r720/pd> (дата обращения: 18.08.2015).
6. NVIDIA GRID – потоковые приложения и игры по запросу. URL: <http://www.nvidia.ru/object/nvidia-grid-ru.html> (дата обращения: 18.08.2015).
7. XenServer – серверная виртуализация и управление гипервизором. URL: <https://www.citrix.ru/products/xenserver/overview.html> (дата обращения: 18.08.2015).
8. Семейство нулевых клиентов Wyse P class для VMware. URL: <http://www.dell.com/ru/business/p/wyse-p-class/pd> (дата обращения: 18.08.2015).
9. PC-over-IP Technology Explained. URL: <http://www.teradici.com/pcoip-technology> (дата обращения: 18.08.2015).
10. Тонкий клиент Wyse серии 7000. URL: <http://www.dell.com/ru/business/p/wyse-z-class/pd> (дата обращения: 18.08.2015).

DOI: 10.15827/0236-235X.112.016-021

Received 09.09.15

THE ANALYSIS OF AN IT INFRASTRUCTURE OF THE ASTRONAUT TRAINING OPEN AUTOMATION SYSTEM TO CREATE AN INTEGRATED SIMULATION COMPLEX

Kovrigin S.N., Head of Department, S.Kovrigin@gctc.ru

(Gagarin Research & Test Cosmonaut Training Center, Star City, Moscow Region, 141160, Russian Federation);

Yanyushkin V.V., Ph.D. (Engineering), Deputy General Manager, v.yanyushkin@simct.ru

(The Center for Space Simulator Development and Personnel Training, Pervomayskaya St. 92, Moscow, 115088, Russian Federation)

Abstract. The article describes testing of astronaut training open automation system at Gagarin Research & Test Cosmonaut Training Center. It includes the development of a test rig for VMware and Citrix virtualization platforms with further deployment of thin clients within a multibranch network. The article provides a short description of the test rig structure and technical solutions including graphics server with NVIDIA GRID K2 cards, two types of virtualization system thin clients and virtualization software. The test object is specific software of two components of astronaut training devices: virtual International Space Station (ISS) desktops and Teleoperator autonomous simulator trainees' desktops featuring motion control handles and navigation. The operation of these two systems is tested in VMware and Citrix. Experiments must confirm that the deployment of simulator virtualization systems and function & simulation stands is possible within the described automation system. The article offers trusted major practical and theoretical solutions as well as a complex of solutions for the development of close software and hardware infrastructure based on the latest technologies. The article clearly defines the main requirements for server nodes, storage systems, networking equipment and target devices. The peculiarity of the work is that all tests were performed on the basis of potential customer's network infrastructure and real workspaces. The work aims at defining whether it is possible to develop a set of integrated solutions at Gagarin Research & Test Cosmonaut Training Center to consolidate function & simulation stands' software within the corporate data center.

Keywords: astronaut training automation system for preparing cosmonauts, integrated complex, data center, network infrastructure, graphics server.

References

1. *Tsentr podgotovki kosmonavtov imeni Yu.A. Gagarina* [Gagarin Research&Test Cosmonaut Training Center]. Available at: <http://www.gctc.ru> (accessed August 18, 2015).
2. *NOU INTUIT. Osnovy kommutatsii* [KNOW INTUIT. Switching Basics]. Available at: <http://www.intuit.ru/studies/courses/3591/833/lecture/14251?page=9> (accessed August 18, 2015).
3. *Virtualizatsiya VMware dlya stolnykh kompyutero, serverov, prilozheny, obshchedostupnykh i gibridnykh oblakov* [VMware virtualization for PC, Servers, Applications, Public and Hybrid Clouds]. Available at: <http://www.vmware.com/ru> (accessed August 18, 2015).
4. *Rezervnoe kopirovanie VMware Sphere* [VMware Sphere Backup]. Available at: <http://www.veeam.com/ru/vmware-esx-backup.html> (accessed August 18, 2015).
5. *Podrobnye svedeniya o servere dlya ustanovki v stoyku PowerEdge R720* [Data on the Server Rack PowerEdge R720]. Available at: <http://www.dell.com/ru/business/p/poweredge-r720/pd> (accessed August 18, 2015).
6. *NVIDIA GRID – potokovye prilozheniya i igry po zaprosu* [NVIDIA GRID – threaded applications and games on request]. Available at: <http://www.nvidia.ru/object/nvidia-grid-ru.html> (accessed 18 August 2015).
7. *XenServer – servernaya virtualizatsiya i upravlenie gipervizorom* [XenServer – Server Virtualization and Hypervisor]. Available at: <https://www.citrix.ru/products/xenserver/overview.html> (accessed August 18, 2015).
8. *Semeystvo nulevykh klientov Wyse P class dlya VMware* [The Family of Zero Clients Wyse P Class for VMware]. Available at: <http://www.dell.com/ru/business/p/wyse-p-class/pd> (accessed August 18, 2015).
9. *PC-over-IP Technology Explained*. Available at: <http://www.teradici.com/pcoip-technology> (accessed August 18, 2015).
10. *Tonky klient Wyse serii 7000* [Wyse 7000 series Thin Client]. Available at: <http://www.dell.com/ru/business/p/wyse-z-class/pd> (accessed August 18, 2015).

УДК 004.72

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.022-027

ОСОБЕННОСТИ МОДЕРНИЗАЦИИ ЦЕНТРА ОБРАБОТКИ ДАННЫХ И КОСМОЦЕНТРА

*А.С. Снытко, инженер 2-й категории, a.snytko@simct.ru;
А.А. Терников, инженер 2-й категории, a.ternikov@simct.ru
(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия)*

В статье рассматриваются использование возможностей современных технологий виртуализации для решения задач консолидации вычислительных ресурсов предприятия в пределах корпоративного центра обработки данных и последующее создание на основе его вычислительных средств единого учебно-тренажерно-моделирующего комплекса в рамках проекта модернизации космоцентра «Астрон» имени космонавта Г.С. Шонина. Описаны развитие, текущее состояние технологий и ключевые преимущества перехода от традиционной модели сетевой инфраструктуры к использованию возможностей серверной виртуализации и виртуализации рабочих станций.

Технология виртуализации рабочих станций предоставляет недоступные ранее возможности динамического распределения ресурсов вычислительного комплекса, обеспечивает отказоустойчивость информационной системы и самое важное для реализации концепции единого интегрированного учебного комплекса – мобильность пользователей за счет виртуализации графических адаптеров и прогрессивных протоколов удаленного подключения к виртуальным рабочим станциям. В проекте использованы решения ведущих мировых производителей программного и аппаратного обеспечения, отвечающие всем требованиям к высокой производительности и надежности.

В результате получена единая среда, отличающаяся высокой степенью автоматизации и упрощенным администрированием. Описаны состав и назначение основных используемых аппаратных компонентов программно-аппаратного комплекса. Приведена структурная схема центра обработки данных Центра тренажеростроения. Рассмотрены задачи, требовавшие решения при модернизации космоцентра «Астрон» и его компонентов, описан процесс модернизации. Приведена структурная схема модернизированного космоцентра. Описаны типовые решения для организации виртуальной среды и особенности применения технологий виртуализации рабочих станций при модернизации космоцентра. Описаны опыт реализации проекта и его результаты для Центра тренажеростроения.

Ключевые слова: центр обработки данных, территориально распределенные организации, учебно-тренажерно-моделирующий комплекс, виртуализация рабочих станций, отказоустойчивость, технологии виртуализации ресурсов, виртуализация GPU, Nvidia GRID, надежность, ИТ-инфраструктура, доступность.

В последние годы широкое распространение получили технологии серверной виртуализации. Они на практике доказали свою эффективность, обеспечили повышенную отказоустойчивость и позволили уменьшить затраты предприятия на аппаратное обеспечение и электроэнергию благодаря консолидации вычислительных ресурсов и средств хранения данных. Кроме того, данные технологии позволили снизить требования к системе охлаждения и сократить площадь помещения серверной. Все это уменьшает совокупную стоимость владения ИТ-инфраструктурой за счет эффективного использования технических средств, например, перераспределения нагрузок для оптимального решения бизнес-задач, а также за счет сокращения расходов на администрирование и повышения качества оказываемых ИТ-услуг. Однако первоначальные затраты на создание центра обработки данных (ЦОД) могут быть очень большими.

Следующим направлением развития технологий виртуализации стала виртуализация рабочих станций, позволившая получить преимущества серверной виртуализации при организации рабочих мест пользователей. В результате развития данного направления ведущие вендоры предоставили возможность виртуализации графических ускорителей, что позволило использовать различные системы автоматизированного проектирова-

ния и трехмерного моделирования в виртуальных средах [1].

Бурное развитие информационных технологий и повышение требований к качеству ИТ-услуг обусловили потребность в модернизации локально-вычислительных сетей (ЛВС) и создании корпоративного ЦОД Центра тренажеростроения и подготовки персонала.

На предприятии осуществляются исследования, разработка, производство и научно-техническое сопровождение эксплуатации технических средств и систем подготовки операторов к управлению сложными динамическими объектами.

К основным целям создания ЦОД предприятия можно отнести следующие:

- необходимость изучения новых информационных технологий, основ виртуализации, современных серверных технологий, операционных систем и нового ПО;
- необходимость организации набора стендов в рамках лаборатории по изучению перспективных серверных технологий, технологий виртуализации ресурсов для отработки принципиально новых технологий при разработке тренажеров, космоцентров и систем обучения;
- осуществление перехода к новым решениям и технологиям на всех этапах работы организации от разработки ПО (тестирование и отладка прило-

жений, создание стендов из виртуализованных выделяемых из ЦОД ресурсов) до развертывания программно-аппаратных комплексов нового поколения у заказчика;

– переход к размещению корпоративных приложений для организации работы Центра тренажеростроения в едином ЦОД (общие БД, веб-службы, серверы хранения, системы обмена и т.д.).

Второй этап – модернизация программно-аппаратного комплекса космоцентра «Астрон» имени космонавта Г.С. Шонина на технических средствах ЦОД [2, 3].

Аппаратной основой ЦОД стало оборудование фирмы DELL: серверы DELL PowerEdge R520 и R720 с графическими ускорителями NVIDIA GRID K2, система хранения данных DELL PowerVault MD3200i, управляемые коммутаторы DELL PowerConnect 6224 и 6248, источники бесперебойного питания APC SmartUPS 3000.

Приведем основные программные компоненты инфраструктуры виртуализации.

- VMware vSphere – платформа виртуализации, в которую входит множество элементов, таких как гипервизоры, система управления ресурсами, центр управления, средства обеспечения отказоустойчивости [4–6]. Обеспечивает функционирование набора виртуальных машин и управление ими, в том числе автоматизированное.

- VMware Horizon View – программный комплекс, обеспечивающий функционирование инфраструктуры виртуальных рабочих станций [7, 8]. Выступает в роли посредника при подключении к виртуальной машине. В его задачи входят создание групп виртуальных машин, управление пользователями и машинами, назначение политик, контроль состояния инфраструктуры. Совместно с VMware vSphere образует целостную среду.

- Citrix XenDesktop – альтернативная платформа инфраструктуры виртуальных рабочих станций. Широко распространена, отличается большим функционалом, гибкими настройками, нетребовательностью к каналам связи. Компания Citrix первая предложила пользователям технологии виртуализации графических адаптеров.

Обеспечение антивирусной защиты в виртуальной среде имеет особенности, связанные с архитектурой виртуальной платформы. Использование стандартного антивируса на множестве виртуальных машин влечет за собой снижение производительности и увеличение объема работ, связанных с управлением защитой. Ведущие производители антивирусных продуктов подготовили специальные решения для виртуальных сред. Платформа VMware vSphere имеет в своем составе специальный компонент – vShield, благодаря которому программа-агент на виртуальной машине не требуется, поэтому для vSphere выпущены специальные версии антивирусов.

Система резервного питания основана на ИБП APC Smart-UPS 3000, оборудованных сетевыми адаптерами. Благодаря интеграции с vSphere обеспечивается автоматическое управление питанием в случае нештатных ситуаций. Структурная схема ЦОД изображена на рисунке 1.

В процессе реализации проекта были успешно решены следующие задачи:

- создана платформа, позволяющая предоставлять необходимые вычислительные ресурсы разработчикам и обеспечивающая функционирование сетевых служб и пользовательских сервисов Центра тренажеростроения: корпоративный почтовый сервис, личные хранилища пользовательской информации, сервис обмена сообщениями между пользователями, сервис системы видеонаблюдения;

- обеспечена отказоустойчивость как на уровне выхода из строя отдельных аппаратных узлов, так и на уровне программных компонентов, таких как СУБД;

- обеспечены создание резервных копий и возможности восстановления в случае возникновения критических ошибок;

- осуществляются автоматический контроль электропитания и автоматическое выключение оборудования в случае возникновения аварийных ситуаций;

- обеспечена антивирусная защита виртуальной среды;

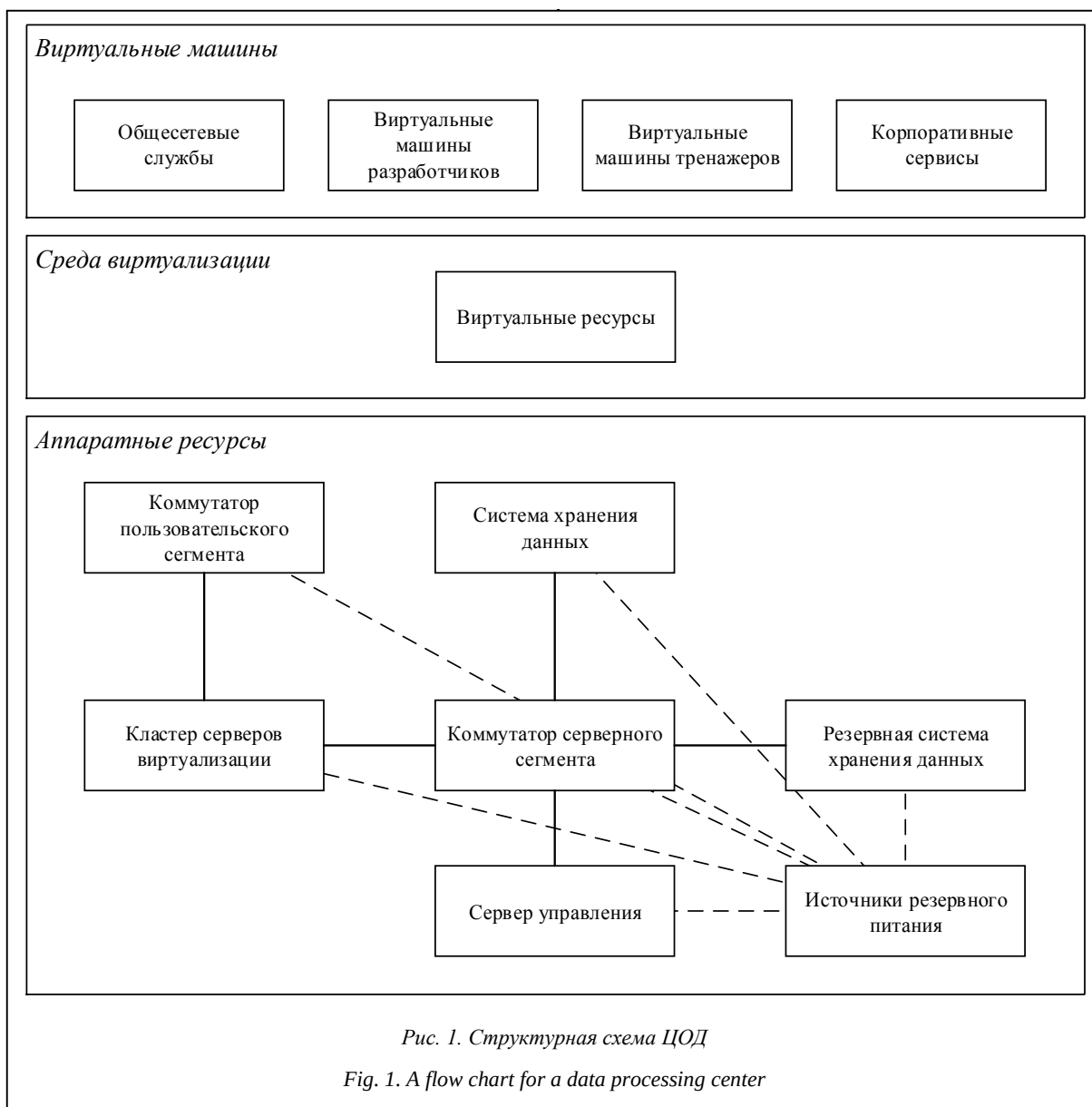
- осуществляется автоматизированное управление ресурсами;

- обеспечена возможность масштабирования системы при возникновении потребности в дополнительных мощностях.

Работоспособность комплекса подтверждена во время эксплуатации и дополнительного тестирования.

По завершении работ первого этапа была спроектирована и внедрена схема модернизации программно-аппаратного комплекса образовательного космоцентра «Астрон», для чего были подробно проанализированы его организационная структура и перспективы применения средств ЦОД, рассмотрены требования к сопровождающим системам космоцентра.

Были рассмотрены основные концепции технологии виртуализации рабочих станций, произведен анализ различных систем управления виртуальной инфраструктурой и разработана схема интеграции такой системы в сеть ЦОД. В качестве сопровождающих систем космоцентра использованы система видео-конференц-связи Openmeetings, система дистанционного обучения Moodle, галерея Resourcespace, система видеонаблюдения Macroscop, система голосовой связи TeamSpeak. Управление виртуальной инфраструктурой организовано средствами ПО VMware Horizon View. Помимо



этого, проведено тестирование альтернативной платформы Citrix XenDesktop, оба решения показали свою высокую эффективность. Для обеспечения функционирования данных подсистем используются ОС Microsoft Windows [9].

Решение Citrix обладает широким функционалом и множеством гибких политик конфигурации, на его основе возможно создание больших распределенных сред. Отличается эффективным протоколом доставки рабочих столов – работа с трехмерной графикой возможна даже через Интернет с шириной канала 2 Мбит/с. Однако решение от VMware показало лучшую производительность и отзывчивость при работе с тренажерами.

Модернизация космического центра проводилась с целью решения ряда проблем:

- необходимость выделения больших площадей под технологические стойки для каждого объекта космического центра;

- невозможность обеспечения мобильности пользователей космического центра;
- большая трудоемкость в обслуживании инфраструктуры;
- повышенное энергопотребление и шумовой фон в помещении.

Структура модернизированного космического центра представлена на рисунке 2.

Появление технологий виртуализации графических адаптеров сделало возможным перенос тренажерных комплексов в виртуальную среду, обеспечивая тем самым возможность создания интегрированного УТМК нового поколения [2, 10]. При этом в силу новизны технологии, ее малой распространенности и нестандартной сферы применения должны быть сделаны соответствующие корректировки при проектировании виртуальной среды.

Существуют типовые конфигурации внедрения, разработанные задолго до появления виртуализации.

лизации графических адаптеров, например, при организации колл-центра могут использоваться наборы связанных клонов *виртуальных машин*

(ВМ), при этом оператору может выделяться случайная машина из уже созданных, генерироваться новая и удаляться по завершении работы. ВМ

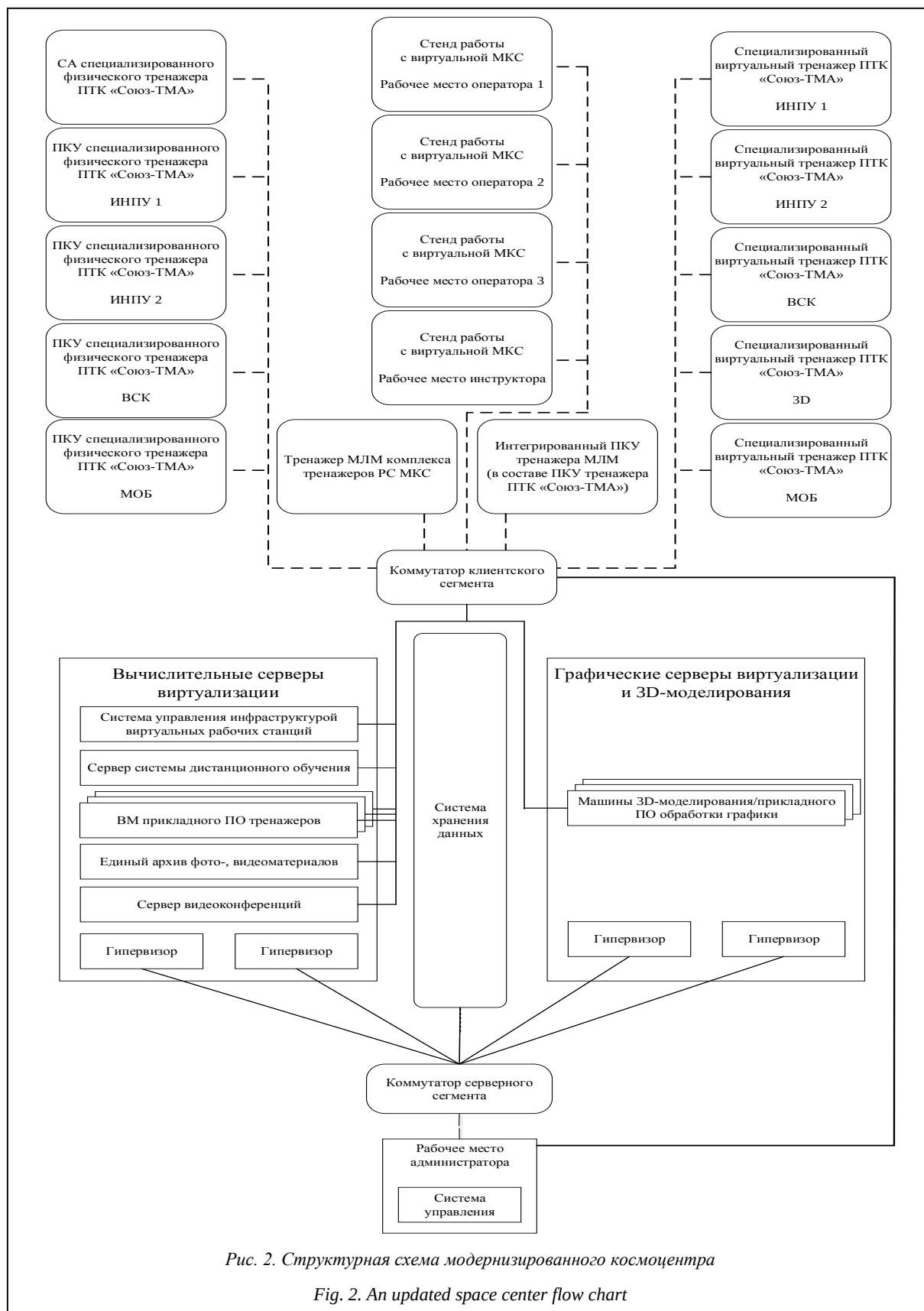


Рис. 2. Структурная схема модернизированного космического центра

Fig. 2. An updated space center flow chart

также может быть закреплена за конкретным оператором. Особенность такой конфигурации в том, что ВМ генерируется на основе базового образа, а все пользовательские данные сохраняются на дельта-диске. Такой подход позволяет значительно экономить дисковое пространство.

При другом подходе на основе единого образа полностью сконфигурированной ВМ создаются полноценные копии, при этом для создания любого количества типовых рабочих мест выполняется минимум административных действий.

Однако в случае с тренажерами эти подходы неприменимы, поскольку каждая ВМ имеет различные задачи и требует разного набора ПО. Кроме того, имеются технические ограничения на реализацию данных подходов в связи с виртуализацией графических адаптеров. Единственная возможность автоматизации – создание базового образа, на основе которого создаются машины тренажеров и в дальнейшем происходят корректировка параметров ВМ, установка и настройка ПО тренажеров. Таким образом, использование виртуализации в тренажерах не позволяет в полной мере применять автоматизированное управление. Ограничением также является невозможность миграции ВМ во время исполнения, что уменьшает гибкость управления ресурсами и ограничивает возможности обеспечения отказоустойчивости.

Технологии графической виртуализации нацелены прежде всего на организацию рабочих мест пользователей систем автоматизированного проектирования и трехмерного моделирования, однако в силу различий в рабочих нагрузках рекомендации по распределению ресурсов для таких сред почти неактуальны при создании тренажеров [8]. Требуемые характеристики ВМ тренажеров были определены экспериментально, и, таким образом, было выработано несколько типовых конфигураций для различных компонентов тренажерных комплексов. Так, для тренажеров виртуальной реальности, таких как «Виртуальная МКС», «Научные Эксперименты», оптимальным оказалось выделение профиля графического процессора GRID K240Q, 6 Гб оперативной памяти и 4 ядра ЦП. Для тренажеров ТМА, МЛМ достаточными являются профиль гра-

фического процессора GRID K220Q, 4 Гб оперативной памяти и 2 ядра ЦП. Помимо этого, потребовалась корректировка параметров протокола доступа к виртуальной рабочей станции.

Интегрированный УТМК создавался с использованием существующих версий ПО тренажеров, при этом были получены хорошие результаты, однако модернизация ПО с ориентацией на виртуализацию позволит оптимизировать потребление ресурсов и повысить эффективность внедрения.

Таким образом, по результатам внедрения и опытной эксплуатации можно говорить об успешной отработке принципов интегрированного УТМК. Применение технологий виртуализации в сфере создания тренажерных комплексов оправданно, они могут использоваться при проектировании решений для заказчиков.

Литература

1. Графическое ускорение для виртуализации десктопов. URL: <http://www.nvidia.ru/object/grid-virtual-desktops-ru.html> (дата обращения: 19.07.2015).
2. Шукшунов В.Е., Янюшкин В.В. Основы разработки и создания интегрированных учебно-тренажерных комплексов для оснащения центров подготовки специалистов по управлению динамическими объектами // Программные продукты и системы. 2013. № 3. С. 3–10.
3. Шукшунов В.Е., Янюшкин В.В. Основы разработки образовательных космоцентров. М.: Машиностроение, 2012. 96 с.
4. Михеев М.О. Администрирование VMware vSphere 5. М.: ДМК-Пресс, 2012. 504 с.
5. Performance Best Practices for VMware vSphere 6.0. URL: <http://www.vmware.com/files/pdf/techpaper/VMware-PerfBestPractices-vSphere6-0.pdf> (дата обращения: 21.08.2015).
6. VMware vSphere 6 Documentation. URL: <https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-6-pubs.html> (дата обращения: 19.08.2015).
7. Лэнгоу Д., Лейбовичи А. Виртуализация настольных компьютеров с помощью VMware View 5. М.: ДМК-Пресс, 2013. 280 с.
8. VMware Horizon 6 with View Performance and Best Practices. URL: <http://www.vmware.com/files/pdf/view/vmware-horizon-view-best-practices-performance-study.pdf> (дата обращения: 21.08.2015).
9. Моримото Р., Ноэл М., Драуби О., Мистри Р., Амарис К. Microsoft Windows Server 2008 R2. Полное руководство; [пер. с англ.]. М.: Вильямс, 2012. 1456 с.
10. Янюшкин В.В. Использование технологий виртуализации вычислительных и графических серверов при проектировании тренажеров, тренажерно-моделирующих комплексов и систем обучения операторов // Программные продукты и системы. 2013. № 3. С. 21–27.

DOI: 10.15827/0236-235X.112.022-027

Received 09.09.15

SPECIAL FEATURES OF DATA-PROCESSING CENTER AND SPACE CENTER UPGRADE

Snytko A.S., 2nd class Engineer, a.snytko@simct.ru;

Ternikov A.A., 2nd class Engineer, a.ternikov@simct.ru

(Don Branch of the Space Simulator Center, Platovsky Ave. 101, Novocherkassk, 346400, Russian Federation)

Abstract. The article describes an advanced virtualization technology which helps to solve many consolidation problems of computing resources within data center. It also considers subsequent creation of the united educational training and simulation system based on computing resources in a modernization project for the space center “Astron” named after cosmonaut Georgiy Shonin. The paper describes the current technology state and development, the main advantages of changing traditional network infrastructure model into server and desktop virtualization. Desktop virtualization technology provides previously unavailable possibility of computing system dynamic distribution, provides information system fault tolerance, and the most important thing for implementing the concept of educational training and simulation complex – it provides user mobility by graphics adapters virtualization and remote advanced protocols to connect virtual desktops. This project uses the solutions

of leading world hardware and software manufacturers, which meet all the requirements to performance and reliability. The result is a united environment with a high degree of automation and simplified administration. The paper describes the composition and application of hardware components in the hardware and software package. The article shows a flow chart describing a data center structure and the quantity of virtual machines in simulators. It also highlights the update process and all problems which are successfully solved in the space center "Astron". The paper describes typical solutions for a virtual environment and special features of virtualization application in space center workstations. The experience of this project and its results can be successfully used in other works and new simulators.

Keywords: data center, geographically dispersed organizations, educational training and simulation complex, desktop virtualization, fail-safety, resource virtualization technology, GPU virtualization, Nvidia GRID, reliability, IT-infrastructure, availability.

References

1. *Accelerated Virtual Desktop*. Available at: [http:// http://www.nvidia.com/object/nvidia-grid.html](http://http://www.nvidia.com/object/nvidia-grid.html) (accessed July 19, 2015).
2. Shukshunov V.E., Yanyushkin V.V. Basics of development and creation of integrated educational and training Complexes for equipping dynamic objects management training centers. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 3, pp. 3–10 (in Russ.).
3. Shukshunov V.E., Yanyushkin V.V. *Osnovy razrabotki obrazovatelnykh kosmotsentrov* [The Basics of Education Space Centers Development]. Moscow, Mashinostroenie Publ., 2012, 96 p.
4. Mikheev M.O. *Administrirovanie VMware vSphere 5* [Mastering VMware vSphere 5]. Moscow, DMK-Press, 2013, 504 p.
5. *Performance Best Practices for VMware vSphere 6.0*. Available at: <http://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf> (accessed August 21, 2015).
6. *VMware vSphere 6 Documentation*. Available at: <https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-6-pubs.html> (accessed August 19, 2015).
7. Langone J., Leibovici A. *VMware View 5 Desktop Virtualization Solutions*. Packt Publ., 2012, 288 p. (Russ. ed.: Moscow, DMK-Press, 2013, 280 p.).
8. *VMware Horizon 6 with View Performance and Best Practices*. Available at: <http://www.vmware.com/files/pdf/view/vmware-horizon-view-best-practices-performance-study.pdf> (accessed August 21, 2015).
9. Morimoto R., Noel M., Droubi O., Mistry R., Amaris C. *Microsoft Windows Server 2008 R2 Unleashed*. Sams Publ., 2010, 1680 p. (Russ. ed.: Moscow, Williams, 2012, 1456 p.).
10. Yanyushkin V.V. Using virtualization technologies of computing and graphics servers when designing complex simulators and operator training systems. *Programmnye produkty i sistemy* [Software & Systems], 2013, no. 3, pp. 21–27.

Реклама

В Центре тренажеростроения и подготовки персонала организован Центр космического образования молодежи

Школьники имеют возможность получить углубленные знания в области математики, физики, астрономии, истории пилотируемой космонавтики. Практические занятия проводятся на интерактивных аналогах космических и авиационных аппаратов (космических тренажерах).

Информация на сайте
www.simct.ru

УДК 004.72

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.028-033

СТЕНД ЭРГОНОМИЧЕСКОЙ ОТРАБОТКИ ПИЛОТИРУЕМОГО ТРАНСПОРТНОГО КОРАБЛЯ КАК СРЕДСТВО ПРОЕКТИРОВАНИЯ И ОТРАБОТКИ ИНТЕРФЕЙСОВ ЭКИПАЖА

М.В. Серов, начальник отдела, космонавт-испытатель, Mark.Serov@rsce.ru;

О.Н. Кукин, к.псих.н., зам. начальника отдела, Oleg.Kukin@rsce.ru

*(Ракетно-космическая корпорация «Энергия» им. С.П. Королева,
ул. Ленина, 4а, г. Королёв, Московская обл., 141070, Россия);*

В.В. Янюшкин, к.т.н., зам. генерального директора, v.yanyushkin@simct.ru

*(Центр тренажеростроения и подготовки персонала,
ул. Первомайская, 92, г. Москва, 115088, Россия);*

В.М. Радченко, зам. директора, v.radchenko@simct.ru;

Р.К. Харагозян, начальник отдела, r.haragozyan@simct.ru

*(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия)*

В статье представлены основные технические и программные решения для стенда эргономической отработки пилотируемого транспортного корабля нового поколения, предназначенного для эргономического проектирования и экспериментальной отработки интерфейсов экипажа с кораблем: варианты дисплейных форматов для информационного обеспечения экипажа, новая ручка управления ориентацией и движением. Инновационная составляющая стенда заключается в возможности итеративного создания и экспериментальной проверки вариантов информационного обеспечения экипажа, а также отработки эргономических характеристик перспективных органов управления, таких как ручка управления ориентацией и движением с полноразмерного макета рабочих мест экипажа. Все работы проводятся параллельно с проектированием и созданием пилотируемого корабля нового поколения. Приводятся функциональный состав стенда и два основных режима работы по подготовке экспериментальных данных и по их апробации на программно-технических средствах. Рассмотрены особенности конструкции стенда, построенной по модульному принципу, что способствует его легкой сборке и разборке, транспортировке, доработке и модернизации с учетом появляющихся новых требований по результатам отработки. Приведены особенности разработки и применения специализированного ПО, позволяющего выполнять ряд задач по удаленному мониторингу и управлению режимами моделирования с места руководителя, используя в том числе беспроводную голосовую связь. Все компоненты объединены во взаимосвязанную систему, обмен данными осуществляется посредством локальной вычислительной сети. В макете применены общепромышленные изделия, доработанные с учетом специфики использования в составе стенда эргономической отработки.

Ключевые слова: пилотируемый транспортный корабль нового поколения, стенд эргономической отработки, информационное обеспечение, интерфейс экипажа, пульт космонавтов, программно-технические средства, среда разработки MasterSCADA.

В настоящее время в России создается *перспективная пилотируемая транспортная система* (ППТС), включающая в себя многоразовый *пилотируемый транспортный корабль* (ПТК) нового поколения, который не только заменит пилотируемые корабли серии «Союз», но и откроет новые возможности, такие как увеличенная грузоподъемность, многоразовость, полеты к Луне [1, 2].

В отличие от ранее принятого подхода к проектированию автоматического пилотируемого корабля [3, 4], приоритет в котором отдается автоматике, а человек осуществляет функции контроля автоматических режимов и выполнения резервных режимов управления, в процессе проектирования ПТК и его составных частей предлагаются итеративное создание и экспериментальная отработка интерфейса экипажа с ПТК, обеспечивающего космонавту активное участие в контроле и управлении кораблем.

Стенд эргономической отработки (СЭО) ПТК, разрабатываемый ракетно-космической корпорацией «Энергия» совместно с Центром тренажеро-

строения и подготовки персонала с 2014 г., предназначен для эргономической оценки, экспериментальной отработки вариантов информационного обеспечения экипажа ПТК, новых *органов управления* (ОУ) и *систем отображения информации* (СОИ), моделирования деятельности экипажа и процессов управления ПТК в наземных условиях. Объект испытаний – интерфейс экипажа с ПТК, включающий экспериментальные и опытные образцы ОУ и СОИ, а также информационная модель систем ПТК в виде набора дисплейных форматов для контроля и управления кораблем.

Программно-аппаратное обеспечение дает возможность подготовки вариантов информационного обеспечения экипажа ПТК с использованием системы разработки информационной модели систем ПТК и экспериментальной отработки интерфейсов экипажа при моделировании вариантов информационного обеспечения ПТК в полноразмерном макете *командного отсека* (КО) корабля.

Система разработки информационных интерфейсов экипажа обеспечивает создание вариантов

дисплейных форматов для представления данных о процессах управления, состоянии и работе бортовых систем, а также о внешней обстановке с целью экспериментальной отработки и эргономической оценки ОУ и СОИ:

- интерактивное создание и изменение информационных кадров с помощью программного пакета разработки;
- создание и пополнение библиотеки графических образов для создания мнемосхем и других элементов информационных кадров;
- настройку связей отображаемой информации с моделями систем и параметрами моделируемых процессов;
- интерактивную настройку параметров информационных моделей;
- отладку информационных интерфейсов экипажа.

Система моделирования предназначена для моделирования процессов взаимодействия экипажа с ПТК в автоматических и ручных режимах управления посредством технических и программных средств и обеспечивает

- имитационное моделирование работы основных бортовых систем ПТК;

– моделирование на *рабочих местах экипажа* (РМЭ) процессов контроля и управления ПТК;

– моделирование внутренней и внешней коммуникации экипажа с использованием имитационной модели бортовой системы связи (между членами экипажа ПТК, с исследователями и операторами – *рабочее место исследователя-оператора* (РМ-ИО));

– управление с РМ-ИО процессами моделирования управления ПТК экипажем, а также внешними связями стенда, отображение информации и управление стендом на РМ-ИО, регистрацию параметров процессов и характеристик деятельности испытуемых ПТК на стенде;

– разработку сценариев процессов моделирования, моделей систем ПТК, БД, ПО контроля и анализа;

– изменение состава и конфигурации РМЭ в соответствии с задачами испытаний.

Структура и состав стенда

Стенд эргономической отработки ПТК состоит из следующих составных частей (рис. 1): полноразмерный макет КО (рис. 2), оснащенный действующим

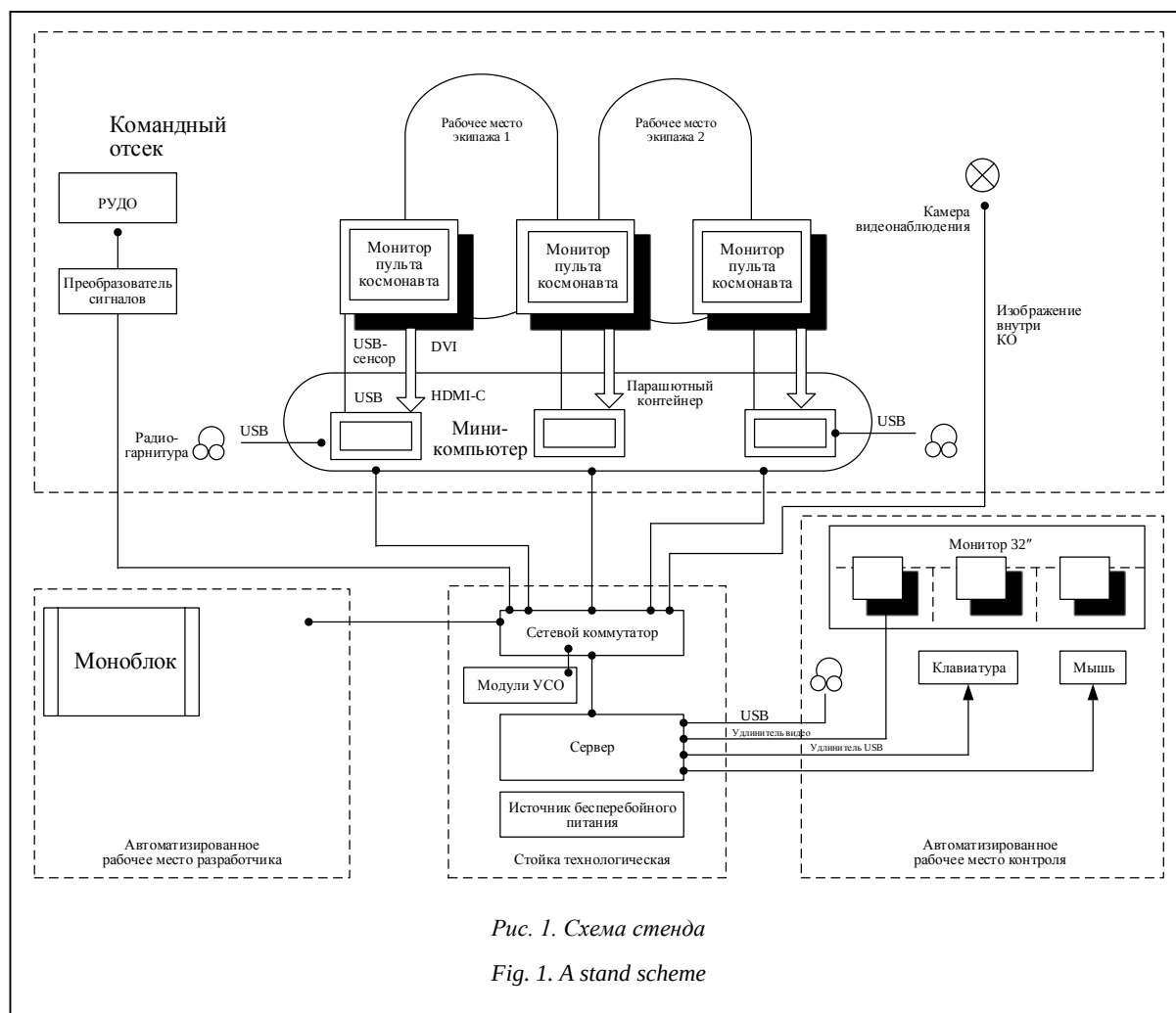


Рис. 1. Схема стенда

Fig. 1. A stand scheme



Рис. 2. Макет командного отсека ПТК

Fig. 2. A training hardware for a hardware and software solution command module

щими макетами пульта космонавтов, кресел космонавтов и другим оборудованием, необходимым для отработки режимов управления и контроля систем ПТК; технологическая стойка для размещения вычислительных средств, сетевого оборудования, системы электропитания и модулей устройств сопряжения (УСО); РМ-ИО, предназначенное для управления работой стенда и контроля проводимых космонавтами операций внутри КО; рабочее место разработчика, предназначенное для создания и корректировки интерфейсов экипажа с ПТК.

Интерьер макета КО представлен на рисунке 3. Макеты кресел позволяют отрабатывать деятельность экипажа на двух основных рабочих местах – РМЭ1, РМЭ2. Пульт космонавта установлен на поворотной раме, которая с помощью механизма может принимать два положения: рабочее и транспортное. На раме размещены три цветных жидкокристаллических сенсорных дисплея с накладными рамками для физической имитации кнопок, расположенных по периферии экрана и обеспечивающих выбор форматов, отображаемых в центральной области экрана. Каждый монитор пульта космонавта подключен к соответствующему ма-



Рис. 3. Интерьер макета КО ПТК и конструктив пульта космонавтов

Fig. 3. A training hardware interior of a hardware and software solution command module and a cosmonaut's panel construct

кету процессорного блока отображения информации, представляющего собой мини-компьютер, обеспечивающий формирование изображения на экране и интерфейс с моделирующей системой. Это обеспечивает три независимых канала управления системами ПТК, ввод команд, мониторинг работы систем и управление отдельными приборами и устройствами с пульта космонавтов.

Действующий макет ручки управления движением и ориентацией (РУДО) объединяет в себе функции двух отдельных органов управления – ручки управления движением (РУД) и ручки управления ориентацией (РУО), имеющихся на существующих транспортных кораблях «Союз-ТМА». РУДО обеспечивает на стенде ручное управление двумя членами экипажа с левого РМЭ и при необходимости с правого РМЭ (левой рукой). РУДО через специализированный преобразователь сигналов, как и мини-компьютеры пульта космонавтов, подсоединены к локальной вычислительной сети стенда ПТК. Это позволяет всем составным компонентам системы обмениваться данными в процессе работы и моделирования.

Беспроводная технологическая связь обеспечивает имитацию системы связи для внутренних переговоров между двумя членами экипажа и взаимодействия с наземными операторами связи.

Комфортные условия работы в командном отсеке обеспечиваются набором вытяжных вентиляторов и светильниками. Контроль (наблюдение) за действиями экипажа с целью оценки их выполнения ведется на РМ-ИО в реальном времени с помощью отдельной IP-камеры.

РМ-ИО оснащено монитором большой диагонали, на котором отображаются состояние каждого из сенсорных мониторов пульта космонавтов, вспомогательная информация о ходе отработки интерфейсов и изображение IP-камеры также для отслеживания выполнения и отработки интерфейсов. Помимо наблюдения, исследователь-оператор может вмешиваться в процесс управления и при необходимости вызывать форматы и управлять системами ПТК: выдавать команды управления, работать с информационными кадрами и форматами, изменять конфигурацию форматов на РМЭ.

ПО стенда

Для работы стенда используются специализированное ПО формирования моделей и форматов, отображаемых на пульте космонавта, на основе коммерческой версии системы Master SCADA; ПО режима захвата экранов пульта космонавтов Lite Manager; ПО технологической связи Team Speak; ПО работы с IP-камерой командного отсека ПТК.

Загрузка системного и прикладного ПО осуществляется на следующих компонентах стенда: АРМ разработчика – после включения загружа-

ются ОС Windows и среда разработки информационных интерфейсов экипажа; АРМ исследователя-оператора – после включения загружаются ОС Windows и системное ПО для выполнения операций захвата экранов пульта космонавтов Lite Manager и организации технологической связи Team Speak, ПО работы с IP-камерой командного отсека ПТК, а также прикладное ПО управления работой стенда; мини-компьютеры, обеспечивающие вывод информации на пульт космонавтов, – после включения загружаются ОС Windows, системное ПО организации технологической связи Team Speak и прикладное ПО отображения форматов Master SCADA [5].

Специализированное ПО формирования моделей и форматов, отображаемых на пульте космонавта, представлено программным продуктом Master SCADA – программным пакетом для проектирования систем диспетчерского управления и сбора данных. Основными свойствами являются модульность, масштабируемость и объектный подход к разработке. Система предназначена для сбора, архивирования, отображения данных, а также для управления различными технологическими процессами. Помимо создания так называемого верхнего уровня, система дает возможность программировать контроллеры с открытой архитектурой. Таким образом, Master SCADA позволяет создавать единый комплексный проект автоматизации [6, 7].

Функционал MasterSCADA может быть расширен за счет использования дополнительных модулей. В базовый функционал входят среда разработки, внутренний архив данных, сообщений и документов, OPC DA- и OPC HDA-клиенты, редактор мнемосхем, редактор отчетов, модуль трендов и модуль журналов, обработка данных, базовые библиотеки функциональных блоков, формирование расписаний и событий. В опциональный набор входят возможность создания сетевого проекта, резервирование, взаимодействие с БД, отраслевые библиотеки функциональных блоков. Проект разрабатывается в единой интегрированной среде (независимо от модульного состава программы). Основным способом создания структуры проекта является установление связей между элементами проекта (объектами, функциональными блоками и переменными) в дереве объектов. При дублировании и копировании связи могут восстанавливаться. При создании проекта могут использоваться стандартные функциональные блоки (ФБ исполнительных механизмов, математические блоки, обработка сигналов и т.д.) либо самостоятельно разработанные пользователем на языках программирования ST, FBD и C#.

Использование MasterSCADA как средства проектирования отработки интерфейсов экипажа обусловлено тем, что применяемое ПО позволяет относительно просто создавать и изменять сцена-

рии моделирования. Для разработки модели системы ПТК используется многофункциональный графический редактор, позволяющий строить мнемосхемы различной сложности с помощью как готовых визуальных элементов из библиотеки, так и новых элементов, создаваемых разработчиком. Логика работы любой системы описывается специализированным скриптом, работа которого обеспечивает моделирование числовых параметров и последующее отображение изменения состояния функциональных блоков на мнемосхемах системы. Такое разделение и наличие достаточно простого пользовательского редактора позволяет при необходимости дорабатывать и изменять получаемые визуальные представления дисплейных форматов в процессе отработки различных эргономических характеристик.

Режим захвата экранов пульта космонавтов обеспечивает программный продукт Lite Manager Free, который обладает основными функциями и позволяет одновременно управлять несколькими компьютерами (до 30) [8]. Преимущества продукта: управление и просмотр удаленного рабочего стола компьютера Remote Desktop, файловый менеджер для работы с файлами и папками, управление питанием компьютера. LiteManager предоставляет понятный и простой интерфейс для удаленного администрирования. Выбираются необходимый режим работы, управление компьютером, передача файлов, просмотр процессов и т.д., осуществляется подключение к удаленному компьютеру. Компьютеры могут отображаться в виде эскизов, на которых видно их текущее состояние. В режиме управления удаленным рабочим столом можно использовать функции Drag&Drop, системный буфер обмена, переключаться между мониторами удаленного компьютера. После автозагрузки ПО и завершения процесса включения стенда выбираются эскизы в главном окне программы удаленных рабочих столов мини-компьютеров пульта космонавтов и выполняется команда их вывода на общий монитор для наблюдения действия и удаленного управления. ПО Lite Manager может быть настроено на дублирование экранов как в 100 %-ном, так и в уменьшенном или увеличенном масштабе.

Специализированное ПО технологической связи Team Speak – программа, предназначенная для голосового общения посредством технологии VoIP [9]. От классического телефона отличается практически неограниченным количеством абонентов, разговаривающих одновременно. ПО состоит из серверной и клиентских частей (серверная часть установлена на АРМ исследователя-оператора, клиентские – на двух мини-компьютерах пульта космонавтов). Для того чтобы пользоваться Team Speak, необходимо знать адрес сервера, на котором установлена серверная часть программы, и пароль, если он требуется для соединения. Для

полноценной работы с Team Speak пользователю требуются наушники или колонки и микрофон – в данном случае используются беспроводные гарнитуры Logitech Wireless Headset Mono H820e. Процесс общения представляет собой конференцию в реальном времени, в которой разговаривающие могут произносить реплики одновременно. Во избежание этого каждый участник может заблокировать лично для себя звук от другого участника. Возможен режим записи всех сообщений в отдельный аудиофайл.

Выбор данного ПО обусловлен тем, что оно полностью реализует все требуемые задачи по созданию циркуляров голосовой связи и отображению дублей экранов пульта космонавта, а также по оперативному изменению параметров и настройке приложений для новых задач на РМ-ИО.

Конструктивные особенности построения макета КО

При проектировании макета ПТК использовались следующие принципы: подобие макета проекту КО по размерам, форме и внутренним объемам, модульность конструкции макета, минимальные весовые характеристики, технологичность сборки и разборки макета [10]. Корпус макета ПТК выполнен из сборно-разборного основания с установленными на нем шестью модулями наружной обшивки. Конструкция основания и наружные обводы корпуса выполнены из алюминиевых тонкостенных прессованных профилей и листового алюминия. На основании макета, кроме ограждающей конструкции, установлены все элементы интерьера: каркасы кресел, кресла, конструкции механизма пульта космонавтов и другие. Они не связаны с конструкцией ограждения, что позволяет в процессе компоновки, монтажа и отработки эргономических характеристик оперативно удалять любой из модулей ограждения, не внося радикальных изменений в конструкцию интерьера КО, оперативно вносить изменения в компоновку интерьера без существенной переделки всей конструкции макета (рис. 4).

Механизм пульта космонавтов в макете КО функционально подобен механизму, созданному в процессе проектно-конструкторской разработки ПТК, и позволяет устанавливать пульт в транспортное и рабочее положение (рис. 5, 6).

Это дает возможность эргономической оценки компоновки рабочих мест космонавтов в КО ПТК и своевременного внесения изменений в конструкцию пульта на этапе выпуска рабочей документации.

Модульное построение макета КО ПТК позволяет вносить изменения в конструкцию макета на стадиях проектно-конструкторской проработки ПТК по результатам эргономической оценки рабочих мест экипажа. В итоге экспериментальной от-

работки интерфейсов экипажа с кораблем вносятся корректировки в проект ПТК.

Представленный стенд эргономической отработки ПТК рассматривается в качестве пилотажного средства, обеспечивающего эффективное взаимодействие экипажа с высокоавтоматизированным космическим кораблем на разных уровнях: контроль и управление отдельными системами, выполнение полетных операций, контроль и оперативное изменение полетных задач и программы полета в целом.

СЭО ПТК должен обеспечить решение этой задачи на базе РКК «Энергия» с использованием но-

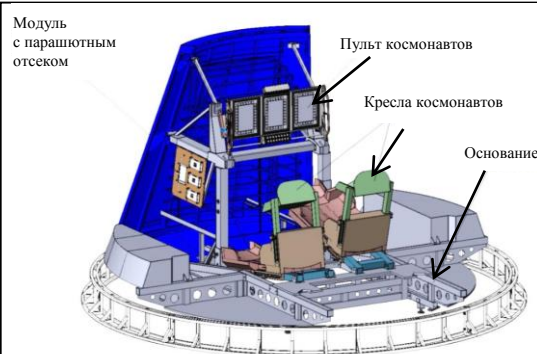


Рис. 4. Внутренняя компоновка макета

Fig. 4. A training hardware internal arrangement

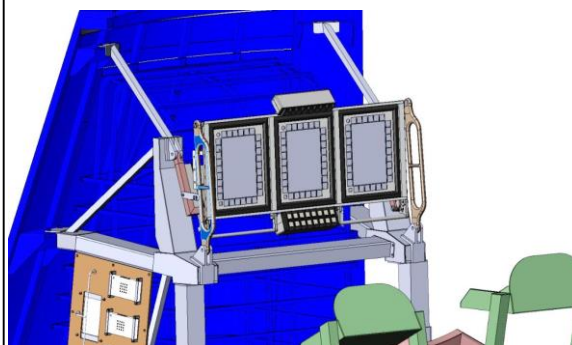


Рис. 5. Пульт космонавта в сложенном положении

Fig. 5. Fold cosmonaut's panel

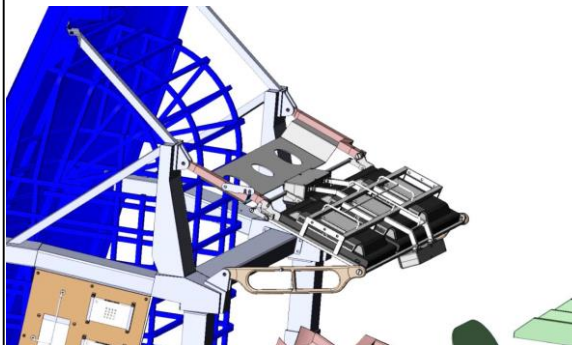


Рис. 6. Пульт космонавта в рабочем положении

Fig. 6. Onposition cosmonaut's panel

вых программных, технических, конструктивных и технологических решений.

Литература

1. Корабль нового поколения (ПТК НП). URL: <http://kosmolenta.com/index.php/new-tech/ptknp> (дата обращения: 20.08.2015).
2. Афанасьев И. Перспективный транспортный корабль нового поколения // Новости космонавтики. 2014. № 09. С. 58–61.
3. Голиков Ю.Я. Современные концепции автоматизации и подходы к человеку и технике // Психол. журнал. 2002. Т. 23. № 1. С. 18–30.
4. Кукин О.Н., Серов М.В. О реализации методов инженерно-психологического проектирования пилотируемых космических аппаратов. Человеческий фактор: проблемы психологии и эргономики. 2013. № 4 (67). С. 34–38.

5. Общая информация о MasterSCADA. URL: http://www.masterscada.ru/?additional_section_id=26 (дата обращения: 20.08.2015).
6. Среда пользовательского программирования и автоматизация проектирования в MasterSCADA. URL: <http://isup.ru/articles/2/763/> (дата обращения: 20.08.2015).
7. MasterSCADA – полнофункциональный пакет программ для разработки АСУ ТП. URL: <http://www.rlda.ru/MasterSCADA.htm> (дата обращения: 20.08.2015).
8. Удаленное управление компьютерами при помощи бесплатной программы LiteManager. URL: <http://www.ixbt.com/soft/litemanager.shtml> (дата обращения: 20.08.2015).
9. Установка и настройка TeamSpeak 3. URL: <https://cfire.mail.ru/forums/showthread.php?t=132893> (дата обращения: 20.08.2015).
10. Шукшунов В.Е., Циблиев В.В., Пототский С.И. Тренажерные комплексы и тренажеры. Технологии разработки и опыт эксплуатации. М.: Машиностроение, 2005. 384 с.

DOI: 10.15827/0236-235X.112.028-033

Received 09.09.15

ERGONOMIC TESTING OF A MANNED TRANSPORT SPACECRAFT FOR CREW INTERFACE DESIGN AND TRAINING

Serov M.V., Head of Department, Test Cosmonaut, Mark.Serov@rsce.ru;

Kukin O.N., Ph.D. (Psychology), Deputy Head of Department, Oleg.Kukin@rsce.ru
(S.P. Korolev Rocket and Space Public Corporation Energia,

Lenina St. 4a, Korolev, Moscow Reg., 141070, Russian Federation);

Yanyushkin V.V., Ph.D. (Engineering), Deputy General Manager, v.yanyushkin@simct.ru
(The Center for Space Simulator Development and Personnel Training,

Pervomayskaya St. 92, Moscow, 115088, Russian Federation);

Radchenko V.M., Deputy Director, v.radchenko@simct.ru;

Kharagozyan R.K., Head of Department, r.haragozyan@simct.ru

(Don Branch of the Space Simulator Center, Platovsky Av. 101, Novocherkassk, 346400, Russian Federation)

Abstract. The article considers basic hardware and software solutions for ergonomic testing of new generation spacecraft. It is supposed to create a stand for ergonomic design and pilot testing of crew interfaces: display formats options to inform the crew, a new joystick to control orientation and movement. The article provides a functional composition of the stand and two main operation modes to prepare experimental data and their validation on the basis of software and hardware platforms. These are the ergonomic features of such perspective control devices as orientation and motion control handle on the one hand and various options of information support for the crew using the latest transforming cosmonaut's console on the other hand that are the most important. All the works are performed along with the development and creation of a unique product. The article describes the main mechanical features of building cosmonauts' workplace that enable easy assembling and disassembling of the product, its transportation, adaptations and update with regards to emerging requirements and conditions of use. The specialized software provides a number of tasks with remote monitoring and management of new generation spacecraft systems from the workplace of the person in charge using wireless voice communications as well. All the components are unified into a correlated system; data exchange is performed via local computing network. The training hardware includes general purpose industrial objects adapted to the use in the ergonomic stand.

Keywords: new generation spacecraft, ergonomic stand, data support, crew interface, cosmonauts' space console, software and hardware environment, MasterSCADA.

References

1. *Korabl novogo pokoleniya (PTK NP)* [New Generation Spacecraft]. Available at: <http://kosmolenta.com/index.php/new-tech/ptknp> (accessed August 20, 2015).
2. Afanasyev I. New generation promising transport spacecraft. *Novosti kosmonavтики* [Space News]. 2014, no. 9, pp. 58–61 (in Russ.).
3. Golikov Yu.Ya. Modern automation concepts and approaches to human and technology. *Psikhol. zhurnal* [Psychol. Journ.]. 2002, vol. 23, no. 1, pp. 18–30 (in Russ.).
4. Kukin O.N., Serov M.V. On the Implementation of the Human Factors Design Methods for Manned Spacecrafts. *Chelovechesky faktor: problemy psikhologii i ergonomiki* [Human Factors: Problems of Psychology and Ergonomics]. 2013, no. 4 (67), pp. 34–38 (in Russ.).
5. *Obshchaya informatsiya o MasterSCADA* [General information about MasterSCADA]. Available at: http://www.masterscada.ru/?additional_section_id=26 (accessed August 20, 2015).
6. *Sreda polzovatel'skogo programmirovaniya i avtomatizatsiya proektirovaniya v MasterSCADA* [Custom programming and design automation in MasterSCADA], available at: <http://isup.ru/articles/2/763/> (accessed August 20, 2015).
7. *MasterSCADA – polnofunktsionalny paket programm dlya razrabotki ASU TP* [MasterSCADA. A Full-Featured Software Package for APCS Development]. Available at: <http://www.rlda.ru/MasterSCADA.htm> (accessed August 20, 2015).
8. *Udalennoe upravlenie kompyuterami pri pomoshchi besplatnoy programmy LiteManager* [PC Remote Control using LiteManager]. Available at: <http://www.ixbt.com/soft/litemanager.shtml> (accessed August 20, 2015).
9. *Ustanovka i nastroyka TeamSpeak 3* [Installation and Configuration of TeamSpeak 3]. Available at: <https://cfire.mail.ru/forums/showthread.php?t=132893> (accessed August 20, 2015).
10. Shukshunov V.E., Tsiibliyev V.V., Pototsky S.I. *Trenazhernye komplekсы i trenazhery. Tekhnologii razrabotki i opyt ekspluatatsii* [Training Complexes and Simulators. Design Technique and Field Experience]. Moscow, Mashinostroenie Publ., 2005, 384 p.

УДК 004.946

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.034-038

ПРИМЕНЕНИЕ СИСТЕМ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ ПРИ ПОДГОТОВКЕ ПЕРСОНАЛА К БОРЬБЕ ЗА ЖИВУЧЕСТЬ

В.М. Радченко, зам. директора, v.radchenko@simct.ru;

Е.М. Бондарь;

А.О. Чуланов, зам. начальника отдела, a.chulanov@simct.ru

*(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия);*

Н.А. Федоров, главный конструктор, nibur@list.ru;

А.Е. Лещина, к.т.н., помощник главного конструктора, nibur@list.ru

*(Центральное конструкторское бюро морской техники «Рубин»,
ул. Марата, 90, г. Санкт-Петербург, 191119, Россия)*

В статье рассматривается проблема подготовки специалистов для обеспечения безопасности жизнедеятельности и осуществления успешной борьбы за живучесть на объектах с повышенной опасностью. Описаны современные тренажерные средства, используемые для подготовки персонала: натурные макеты отраслевых учебных центров, комплексные и специализированные тренажеры. Рассматриваются способы организации рабочего места обучаемого на основе реальных макетов или компьютерной имитации оборудования, а также применение систем виртуальной реальности в современных средствах подготовки и предлагается использование системы виртуальной реальности в подготовке специалистов к борьбе за живучесть. Приводится описание автоматизированного процесса обучения операциям по борьбе за живучесть на основе систем виртуальной реальности и широкого имитационного моделирования процессов возникновения и развития аварий. Также описываются виртуальная среда, взаимосвязь с имитационной моделью оборудования, способы взаимодействия обучаемого с виртуальной средой при отработке навыков локализации и ликвидации аварийных ситуаций.

Ключевые слова: борьба за живучесть, подготовка персонала, тренажер, рабочее место обучаемого, система виртуальной реальности, имитационное моделирование, аварийная ситуация.

При эксплуатации сложных инженерных объектов, таких как нефтегазодобывающие платформы, корабли и подводные аппараты, особенно важно обеспечить безопасность их жизнедеятельности. Мировой опыт подсказывает, что даже при применении высоконадежных современных технических средств и технологических регламентов на данных объектах периодически возникают аварии, вызванные недостатками в проектировании и при постройке, отказами устройств управления [1], человеческим фактором [2]. Совершенствование навыков по использованию разнообразных спасательных и технических средств по борьбе с аварийными ситуациями осуществляется в специализированных учебных центрах.

Для успешной борьбы за живучесть (БЗЖ), представляющей комплекс мер, направленных на предотвращение возникновения и развития аварийных ситуаций, необходима подготовка специалистов в вопросах эвакуации с аварийного объекта, использования индивидуальных и коллективных спасательных средств и средств защиты, способов локализации и ликвидации пожаров, борьбы с затоплением и другими аварийными ситуациями.

Тренажеры БЗЖ

Для подготовки специалистов к осуществлению БЗЖ применяется полный спектр современных средств подготовки, важное место среди которых занимают тренажеры. Использование тренажеров обусловлено целым рядом их преимуществ

по сравнению с другими средствами, например, меньшая стоимость эксплуатации, более высокая педагогическая эффективность [3], возможность отработки реальных навыков и умений.

Тренажер может быть выполнен в виде натурального макета, обеспечивающего создание опасных условий с помощью различных технических средств. Обычно натуральный макет размещается в отраслевых специализированных учебных центрах подготовки персонала [4, 5]. Натурные макеты занимают большие пространства, но именно с их помощью обучаемый помещается непосредственно в опасную среду, где борется с реальным пожаром различными средствами пожаротушения, выполняет действия по заделыванию пробоин и борьбе с поступающей водой, облачается в спасательное снаряжение и выполняет операции на глубине.

В современных тренажерах широкое применение находят новые информационные технологии и средства вычислительной техники. Базовым в них является понятие *рабочего места обучаемого* (РМО), обеспечивающего развитие перцептивных навыков обучаемых. РМО может быть создано в различных вариантах, например в физическом исполнении, полностью соответствующем реальному пульту управления штатной системы, либо в виде высококачественной имитации внешнего вида приборов и устройств на экране монитора ПЭВМ [3, 6].

Внимание обучаемого замыкается на средствах РМО, а все остальное окружение, имеющее место в реальном мире, подменяется информацией, со-

здаваемой средствами вычислительной техники на основе имитационных моделей [2].

Эволюцией предложенной концепции РМО является переход от формирования перцептивных навыков на базе внешних образов к развитию абстрактного интеллектуального мышления на базе общего представления об устройстве обслуживаемых систем. Развитие интеллектуальных навыков осуществляется путем предоставления обучаемому информации о системе в виде блок-схемы. Данный подход нашел свое отражение в комплексном тренажере подготовки к борьбе за живучесть корабля [7], в котором на базе локальной вычислительной сети и комплекса ПЭВМ обеспечивается подготовка экипажей кораблей.

Комплексный тренажер объединяет в себе двумерное представление состояния оборудования бортовых систем с адекватной имитационной моделью функционирования, на базе которых возможна отработка аварийных ситуаций, связанных с возникновением и развитием пожара, поступлением забортной воды и прорывом труб системы воздуха высокого давления. Ситуационное моделирование развития аварии характеризуется общей непредсказуемостью, что погружает обучаемого в состояние эмоциональной напряженности [3], заставляет принимать решения о применении средств БЗЖ в условиях дефицита времени. По результатам работы обучаемого на основе адекватной имитационной модели дается ответ на вопрос о своевременности и эффективности выполненных действий, выражающийся в прекращении развития аварийной ситуации или переходе ее в более усугубленную фазу [8].

В последнее время разработчики тренажерной техники особое внимание уделяют *системам виртуальной реальности* (СВР), в использовании которых видится решение многих проблем тренажеростроения [2, 6]. Например, фирма Kongsberg, являющаяся одним из ведущих мировых производителей тренажерных систем, на базе ПЭВМ в виртуальной форме смоделировала машинное отделение реального гражданского судна. Система позволяет поворачивать маховики, считывать показания измерительных приборов, перемещаться в смежные помещения, выполнять иные действия, предусмотренные инструкцией по управлению техническими устройствами. Другим примером использования СВР является компьютерный интерактивный тренажер по обслуживанию корабельного оборудования [9], в котором средствами 3D-графики представляется рабочее пространство с расположенным в нем оборудованием. Стены помещений сделаны с переменной прозрачностью, что позволяет получить обзор всего пространства и обеспечить отработку навыков по обслуживанию бортовых систем корабля.

В статье предлагается реализация автоматизированного РМО в составе комплексных и специа-

лизированных тренажеров по обеспечению БЗЖ на базе СВР [10]. Такой подход позволит соединить в одном тренажерном средстве обучение способам борьбы с аварийными ситуациями и управлению бортовым оборудованием в условиях развития аварии. Предлагается с помощью современных средств ввода управляющих воздействий устранить такой недостаток современных РМО на базе СВР, как слабая или интуитивно неясная интерактивность с объектами виртуальной среды, а также расширить использование СВР на все бортовое оборудование управляемого объекта и снять ограничение области рабочего пространства одним помещением, позволив отрабатывать такие элементы тренировки, как изменяющиеся рубежи обороны при борьбе с аварийной ситуацией.

Автоматизированное обучение БЗЖ

Основная суть заключается в объединении механизмов развития перцептивных и интеллектуальных навыков. Обучаемому средствами интерактивной трехмерной графики дается визуальная информация о состоянии окружающего оборудования. В качестве РМО могут выступать любые замкнутые и открытые пространства произвольных размеров и конфигурации, поскольку возможности СВР не ограничивают в этом разработчиков тренажера. Вопрос может стоять только в точности и качестве моделирования окружающего пространства, а оно задается с высокой степенью соответствия реальному рабочему пространству. Оборудование виртуального окружения имеет органы индикации и управления, такие как кнопки, тумблеры, регуляторы [11], которые отображают состояние моделируемого оборудования согласно данным имитационной модели (рис. 1).



Рис. 1. Работа с переносными приборами газового контроля в виртуальных помещениях объекта

Fig. 1. Working with portable gas checking devices in virtual rooms

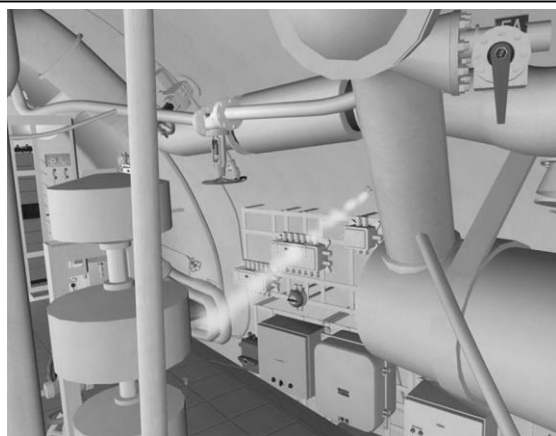


Рис. 2. Вид бьющей из пробоины струи воды (слева) и задымление пространства с пламенем на приборных панелях (справа)

Fig. 2. The view of the spraying jet in a shot hole (on the left) and smoke coverage with flames on a control panel (on the right)

В виртуальной среде обучаемый может осуществлять управляющие воздействия на окружающее оборудование и получать адекватную реакцию на свои воздействия в виде изменения состояния органов управления. Интерфейс человеко-машинного взаимодействия зависит от уровня реализации СВР, может базироваться на стандартных средствах ввода, таких как клавиатура, мышь, джойстик, или иметь более сложные средства виртуального взаимодействия, такие как перчатка данных с СВР в стереорежиме [12].

Виртуальная среда обеспечивает воспроизведение акустической обстановки и моделирование визуальных эффектов физических явлений, таких как дым, пламя, бьющие струи воды и водная поверхность [13]. Проявление тех или иных явлений осуществляется на основании данных имитационной модели развития аварийной ситуации, которая по команде руководителя тренировки отрабатывает требуемый план возникновения и развития аварии. При достижении имитационной моделью некоторых критических значений в виртуальном пространстве начинают проявляться адекватные развивающейся аварийной ситуации аудиовизуальные образы. Например, появляется шум поступающей забортной воды от бьющей из пробоины струи воды или задымление окружающего пространства с появлением открытого пламени на горящем оборудовании (рис. 2).

Имитационная модель развития аварийной ситуации учитывает влияние вредных факторов развития аварии на виртуальную сущность обучаемого и рассчитывает наносимый ущерб здоровью. Проявление воздействий опасных факторов выражается в демонстрации обучаемому соответствующих аудиовизуальных образов. Например, при удусье продуктами горения система воспроизводит биение в звуковом и потемнение в визуальном канале ощущений, а при переохлаждении имити-

руется дрожь визуального канала целиком. Таким образом, поиск и применение средств индивидуальной или групповой защиты при отработке задач БЗЖ является одним из главных факторов выживания и успешного завершения мероприятий по ликвидации аварии.

На основе данных, поступающих из имитационной модели развития аварийной ситуации, СВР воспроизводит аудиовизуальные образы предаварийной и аварийной ситуаций. Задачами обучаемого являются поиск источника аварии, правильная оценка обстановки, применение средств индивидуальной защиты, выбор способа локализации и ликвидации аварии, принятие решения об эвакуации.

На основе принятого в используемой СВР механизма человеко-машинного взаимодействия обучаемый выполняет операции по заделыванию пробоины и ликвидации возгорания выбранным им средством (рис. 3).

Имитационная модель развития аварии принимает информацию о действиях обучаемого, изменяет параметры моделируемой аварии, замедляя процесс развития в случае успешных действий или продолжая его в противном случае. По результатам тренировки руководитель может оценить деятельность обучаемого в виртуальной среде по общему признаку ликвидации им возникшей аварии или по детальному протоколу действий, совершенных за время тренировки.

Применение автоматизированного обучения персонала на основе использования 3D-графики [10] позволяет привить обучаемому первоначальные практические навыки по ликвидации последствий аварий на рабочем месте, закрепить в его сознании последовательность действий, необходимых для обеспечения собственной безопасности и безопасности технологических объектов, на которых он должен осуществлять свою деятельность.



Рис. 3. Применение раздвижного упора для ликвидации пробоя (слева)
и использование пенного огнетушителя при ликвидации возгорания (справа)

Fig. 3. Using telescope prop to remove breakdown (on the left) and using foam fire fighter to remove flame formation (on the right)

Подытоживая, отметим, что в статье дается обзор современных средств обеспечения подготовки персонала к БЗЖ и предложен вариант использования РМО на основе 3D-графики, который был реализован в составе комплексного тренажера для обучения экипажа корабля по заданию ЦКБ МТ «Рубин» (г. Санкт-Петербург).

Литература

1. Караев Р.Н., Разуваев В.Н. Безопасность жизнедеятельности, аварийно-спасательная и противопожарная подготовка персонала нефтяных платформ // Морской вестн. 2010. № 1 (33). С. 47–50.
2. Ковтун Л.И., Поляков В.Н., Семенов С.В. Человеческий фактор и техногенные аварии на кораблях // Судостроение. 2003. № 5. С. 11–16.
3. Шилов К.Ю., Кобзев В.В. Интеллектуальные тренажеры корабельных операторов // Морской вестн. 2009. № 1 (29). С. 41–45.
4. Васюков В. Состояние и пути развития учебно-тренировочных средств ВМФ // В кн.: Оружие и технологии России. XXI век. Т. XVIII. Тренажеры и технические средства обучения, М.: Оружие и технологии, 2009. С. 380–387.
5. Учебно-тренировочный комплекс по борьбе с водой и пожаром. В кн.: Оружие и технологии России. XXI век. Т. XVIII. Тренажеры и технические средства обучения, М.: Оружие и технологии, 2009. С. 453–455.
6. Шилов К.Ю., Кобзев В.В., Сизов Ю.Н. Интеллектуальные тренажеры ОАО «Концерн «НПО «Аврора» // Морской вестн. 2012. № 4 (44). С. 67–71.
7. Комплексный тренажер подготовки к борьбе за живучесть корабля. В кн.: Оружие и технологии России. XXI век. Т. XVIII. Тренажеры и технические средства обучения, М.: Оружие и технологии, 2009. С. 428–429.
8. Третьяков А.В., Колесник В.А. Моделирование развития пожароопасной ситуации в корабельном отсеке на основе методов системного анализа с использованием аппарата структурных матриц // Морской вестн. 2011. № 3 (39). С. 81.
9. Компьютерные интерактивные тренажеры по обслуживанию корабельного оборудования. В кн.: Оружие и технологии России. XXI век. Т. XVIII. Тренажеры и технические средства обучения, М.: Оружие и технологии, 2009. С. 503–505.
10. Бирюков Ю.Б., Бондарь Е.М., Глазко Ю.Г., Радченко В.М., Сединко А.М., Чуланов А.О. Способ автоматизированного обучения персонала морских нефтегазодобывающих платформ действиям в экстремальных и аварийных условиях: пат. 2455699. Рос. Федерация № 2010145796/12; заявл. 11.11.2010; опубл. 10.07.2012. Бюл. № 19. 8 с.
11. Михайлюк М.В., Хураськин И.А. Реализация виртуальных пультов в тренажерных системах // Пилотируемые полеты в космос: сб. тез. докл. Междунар. науч.-практич. конф. Секц. 4. Технические средства и технологии для построения тренажеров. Звездный городок, 2007. С. 26–28.
12. Михайлюк М.В., Фомичев В.М., Хураськин И.А. Технология взаимодействия человека с виртуальной средой в стереорежиме // Технические средства и технологии для построения тренажеров: матер. науч.-технич. сем. Звездный городок, 2004. Вып. 5. С. 20–22.
13. Радченко В.М., Бондарь Е.М., Чуланов А.О., Шабуров Д.В. Технология создания компьютерных систем обучения персонала морских сооружений по обеспечению безопасности и действиям в экстремальных и аварийных условиях на базе использования интерактивной трехмерной графики // RAO / CIS Offshore 2011: тр. 10-й Междунар. конф. СПб: Химиздат, 2011.

DOI: 10.15827/0236-235X.112.034-038

Received 09.09.15

VIRTUAL REALITY SYSTEMS IN DAMAGE CONTROL TRAINING OF PERSONNEL

Radchenko V.M., Deputy Director, v.radchenko@simct.ru;

Bondar E.M.;

Chulanov A.O., Deputy Head of Department, a.chulanov@simct.ru

(Don Branch of the Space Simulator Center, Platovsky Av. 101, Novocherkassk, 346400, Russian Federation);

Fedorov N.A., Chief Designer, nibur@list.ru;
Leshchina A.E., Ph.D. (Engineering), Assistant Chief Designer, nibur@list.ru
(Central Design Bureau for Marine Engineering "Rubin",
Marata St. 90, St. Petersburg, 191119, Russian Federation)

Abstract. The article considers the problem of specialists training to ensure safety of life and for successful damage control in higher risk places. It also describes modern simulator tools for personnel training, such as natural layouts in field training centers, complex and part-task simulators. The article considers the ways of operator workplace organization based on real layouts or computer simulation of equipment. The paper reviews virtual reality systems applicability in modern training tools and offers using virtual reality systems to train specialists in damage control. The article gives the description of the automated training process of damage control based on virtual reality systems and wide simulation modeling of accident appearance and progression. The paper also describes virtual environment, its communication with equipment simulation model, ways of trainee interaction with virtual environment in skill training regarding localization and liquidation of emergency situations.

Keywords: damage control, personnel training, simulator, trainee's workplace, virtual reality system, simulation modeling, emergency situation.

References

1. Karaev R.N., Razuvaev V.N. Oil Platform Safe Operation, Staff Rescue and Fire Control Training. *Morskoy vestnik*. 2010, no. 1 (33), pp. 47–50 (in Russ.).
2. Kovtun L.I., Polyakov V.N., Semenov S.V. Human factor and technological accidents on shipboard. *Sudostroenie* [Shipbuilding]. 2003, no. 5, pp. 11–16 (in Russ.).
3. Shilov K.Yu., Kobzev V.V. Intellectual trainers for ship operators. *Morskoy vestnik*. 2009, no. 1 (29), pp. 41–45 (in Russ.).
4. Vasyukov V. Satus and development paths of the Navy's training aids. *Oruzhie i tekhnologii Rossii. XXI vek. Tom XVIII. Trenazhery i tekhnicheskie sredstva obucheniya* [Encyclopedia "Russia's Arms and Technologies. The XXI Century Encyclopedia". Vol. 18 "Trainers and Training Facilities"]. Moscow, Oruzhie i tekhnologii Publ., 2009, pp. 380–387 (in Russ.).
5. Water and fire control training system. *Oruzhie i tekhnologii Rossii. XXI vek. Tom XVIII. Trenazhery i tekhnicheskie sredstva obucheniya* [Encyclopedia "Russia's Arms and Technologies. The XXI Century Encyclopedia". vol. 18 "Trainers and Training Facilities"]. Moscow, Oruzhie i tekhnologii Publ., 2009, pp. 453–455 (in Russ.).
6. Shilov K.Yu., Kobzev V.V., Sizov Yu.N. Mental training simulators of OJSC "Concern "Avrora" Scientific and Production Company". *Morskoy vestnik*. 2012, no. 4 (44), pp. 67–71 (in Russ.).
7. Integrated ship damage control trainer. *Oruzhie i tekhnologii Rossii. XXI vek. Tom XVIII. Trenazhery i tekhnicheskie sredstva obucheniya* [Encyclopedia "Russia's Arms and Technologies. The XXI Century Encyclopedia". vol. 18 "Trainers and Training Facilities"]. Moscow, Oruzhie i tekhnologii Publ., 2009, pp. 428–429 (in Russ.).
8. Tretyakov A.V., Kolesnik V.A. Modeling the development of fire hazardous situation in ship's compartment on the basis of systematic analysis methods using the apparatus of the structural matrices. *Morskoy vestnik*. 2011, no. 3 (39), pp. 81–86 (in Russ.).
9. Ship equipment maintenance interactive computer trainers. *Oruzhie i tekhnologii Rossii. XXI vek. Tom XVIII. Trenazhery i tekhnicheskie sredstva obucheniya* [Encyclopedia "Russia's Arms and Technologies. The XXI Century Encyclopedia". vol. 18 "Trainers and Training Facilities"]. Moscow, Oruzhie i tekhnologii Publ., 2009, pp. 503–505 (in Russ.).
10. Radchenko V.M., Glazko Yu.G., Chulanov A.O. *Sposob avtomatizirovannogo obucheniya personala morskikh neftegazodobyvayushchikh platform deystviyam v ekstremalnykh i avariynnykh usloviyakh* [Personnel computer-aided learning method in sea and gas and oil producing platforms to act under extreme and emergency conditions]. Patent PF, no. 2455699, 2010 (in Russ.).
11. Mikhaylyuk M.V., Khuraskin I.A. Virtual control panels implementation in simulator systems. *Pilotiruemye polety v kosmos: sb. tez. dokl. Mezhdunar. nauch.-praktich. konf.* [Space Manned Missions: Proc. of the Int. Science and Practice Conf.]. Star City, 2007, pp. 26–28 (in Russ.).
12. Mikhaylyuk M.V., Fomichev V.M., Khuraskin I.A. A technology of man and virtual environment interaction in a stereo mode. *Tekhnicheskie sredstva i tekhnologii dlya postroeniya trenazherov. Mater. nauch.-tekhn. seminara* [Hardware and Technologies to Construct Simulators. Proc. of the Int. Science and Practice Workshop]. Star City, 2004, no. 5, pp. 20–22 (in Russ.).
13. Radchenko V.M., Bondar E.M., Chulanov A.O., Shaburov D.V. Computer-based training systems for offshore structures personnel for safety and actions in emergency situations based on interactive 3D graphics. *Trudy 10 Mezhdunar. konf. RAO/CIS Offshore 2011* [Proc. of the Int. Conf. RAO/CIS Offshore 2011]. SPb, Khimizdat Publ., 2011.

УДК 004.588, 004.823

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.039-044

АКТИВНОЕ ИНФОРМАЦИОННОЕ СОПРОВОЖДЕНИЕ ДЕЙСТВИЙ ЭКИПАЖА

С.И. Кравченко, к.т.н., начальник сектора, artstory@bk.ru;

А.Г. Душенко, к.т.н., начальник отдела;

Д.А. Андреев, инженер-программист, dimak134@yandex.ru

*(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия);*

А.Г. Котов, начальник отдела, alexander.kotov@sfo.ru

*(Ракетно-космическая корпорация «Энергия» им. С.П. Королева,
ул. Ленина, 4а, г. Королев, Московская обл., 141070, Россия)*

В статье отмечается, что задачи информационной поддержки деятельности экипажей космических кораблей и станций непрерывно расширяются за счет насыщенности бортовых планов, увеличения доли научной деятельности в бортовом расписании; возрастают требования к оперативности и наглядности бортовой документации, большое значение приобретают мультимедийный иллюстративный материал, дополнительная справочная информация, звуковое сопровождение. Приводятся результаты экспериментальной эксплуатации на борту Международной космической станции прототипа многофункциональной информационной системы, в ходе которой были определены первоочередные интерактивные объекты и функции, необходимые для обеспечения полноценной и эффективной работы экипажа: возможность сенсорного воздействия на модель окружающей среды, изменение вида, позиций и внутренних параметров виртуальных объектов, ввод данных в интерактивные области документов, установка отметок о выполнении предписанных действий, исполнение предписанных временных интервалов, работа с системой авторизованных закладок, участие оператора в анкетировании по результатам завершённой рабочей сессии, протоколирование поведения космонавта в информационной среде и передача протоколов на Землю.

Формулируются проблемы развития бортовых информационных систем и обеспечивающих их наземных комплексов. Затрагивается тема расширения возможностей интерактивного взаимодействия оператора и информационной среды.

Предлагаются ключевые решения, описываются новые средства подготовки документации. Приводятся примеры внедрения в среду активных элементов. Уделяется внимание ситуационному поведению оператора, работе с виртуальными органами управления, протоколированию результатов и оценке деятельности экипажа. Дается описание специального редактора управляющего слоя электронного документа. Предлагаются решения для преодоления неопределенности выбора операционной системы и технической платформы персональных планшетов.

Ключевые слова: бортовая документация, модуль интерфейса, мультимедиа, Российский сегмент МКС, интерактивные зоны.

Информационная поддержка деятельности экипажей космических кораблей и станций расширяется за счет повышения насыщенности бортовых планов, увеличения доли научной деятельности в общем расписании [1]. Соответственно возрастают требования к оперативности предъявления и наглядности бортовой документации, большое значение приобретают мультимедийный иллюстративный материал, дополнительная справочная информация, звуковое сопровождение [2].

Увеличивается информационная нагрузка на оператора, изменяется качественный характер его работы в информационном пространстве. Особую роль приобретают интерактивные составляющие бортовой документации. Оператор теперь не только работает на прием сведений, но и активно общается с информационной системой [3].

В результате разработки и экспериментальной эксплуатации на борту *Международной космической станции* (МКС) прототипа *многофункциональной информационной системы* (МИС) [4] были определены следующие интерактивные объекты и функции, необходимые для обеспечения полноценной и эффективной работы экипажа:

- возможность сенсорного воздействия на модель окружающей среды, изменение вида, позиций и внутренних параметров виртуальных объектов [5];

- ввод данных в интерактивные области документов *электронной бортовой документации* (ЭБД) (формуляры, аргументы функций, записи в блокнотах);

- установка отметок о выполнении предписанных действий;

- исполнение предписанных временных интервалов для отложенных действий с последующей реакцией;

- работа с системой авторизованных закладок;

- участие оператора в анкетировании по результатам завершённой рабочей сессии;

- протоколирование поведения космонавта в информационной среде и передача протоколов на Землю, офисная обработка протоколов, автоматизированная оценка действий экипажа.

Для реализации перечисленных требований в интерфейс и алгоритмы информационной системы должны быть внедрены соответствующие интерак-

тивные элементы управления [6] и средства привязки мультимедиа [7].

Существует проблема совместимости этих элементов с традиционными редакторами ЭБД.

Еще один важнейший аспект работы МИС – ее пространственная распределенность. Подготовка информации осуществляется наземным комплексом, преобразование этой информации в доступную оператору форму – на борту космического объекта.

Объемы обновляемой информации также имеют критическое значение для производительности существующей системы передачи данных по каналу Земля–борт. Проблема обостряется из-за отсутствия постоянного канала. Связь действует периодически в привязке орбиты к наземным пунктам. В этом отношении ключевой становится методика обновления информации в бортовой системе.

Дополнительной проблемой является неопределенность выбора ОС бортовых вычислительных средств. Наличие терминалов в планшетной реализации с различными техническими и операционными платформами создает дополнительные трудности.

Были проведены исследования с целью выбора оптимальной методологии работы наземного и бортового комплексов МИС и преодоления перечисленных проблем. В работе представлены их основные результаты.

Внедрение в ЭБД мультимедийных и интерактивных элементов

Основное различие классической, «бумажной» БД и мультимедийной интерактивной ЭБД состоит в постоянном расширении функциональных и пользовательских возможностей последней. Условная граница между БД и ЭБД проходит по линии разделения отображаемых элементов на две категории:

- визуальная основа процедуры (страницы), соответствующая стандарту ODF и имеющая привычный для космонавта вид; это то, что можно напечатать на бумаге; визуальная основа консервативна и не изменяется в ходе работы с документацией;

- интерактивный (управляющий) слой, состоящий из виртуальных органов управления ЭБД: кнопок, подсветки пунктов, отметок об исполнении, зон для генерации заполняемых формуляров, указателей расшифровки аббревиатур, гиперссылок, значков запуска мультимедийных приложений, привязки дикторских комментариев, записей в авторизованном блокноте и др.; эти элементы, естественно, не могут быть напечатаны на бумаге, они изменяются, трансформируются в соответствии со сценарием действий экипажа и бортовыми планами.

С учетом непрерывного совершенствования офисных редакторских инструментов [8] и не зависящей от них эволюции интерактивных средств космического назначения информационная среда разделена на две категории: визуальную основу и интерактивный управляющий слой. Их параллельная реализация позволяет, с одной стороны, использовать различные редакторские инструменты и не зависеть от закрытого кода редактора, с другой – развивать интерактивную составляющую в условиях непрерывно меняющихся требований прикладной среды.

Для наземной разработки интерактивных (управляющих) XML-файлов был создан простой и эффективный *визуальный редактор* (ВР) класса WISIWIG (рис. 1).

ВР устанавливается на АРМ разработчика ЭБД. В левой части экрана ВР выводится визуальная основа страницы ЭБД, в правой – так называемый *Мастер интерактивного слоя* (далее – Мастер).

Две верхние строки Мастера отображают адреса создаваемого интерактивного слоя и корреспондируемого файла визуальной основы.

Для взаимодействия с внешней средой на панели редактора имеется группа кнопок со следующими функциями:

- поиск и выбор файла визуальной основы;
- запись результата редактирования в файловую структуру *модуля интерфейса пользователя* (МИП);
- свертка окна редактора на нижнюю панель Windows;
- свертка окна в каскад окон Windows;
- выход из программы;
- откат в предыдущее состояние (отмена последнего действия);
- добавление новой зоны;
- выделение всех имеющихся на странице зон;
- снятие выделения со всех зон;
- удаление группы выделенных зон;
- удаление всех зон (очистка страницы);
- блокировка всех зон, функционал которых отличается от функционала фокусной зоны;
- выравнивание позиции группы выделенных зон по левому краю;
- уравнивание ширины выделенных зон по самой широкой зоне группы;
- выравнивание позиции группы выделенных зон по верхнему краю;
- уравнивание высоты выделенных зон по самой высокой зоне группы;
- копирование выделенных зон в буфер;
- вставка зон из буфера в позицию специального курсора, который появляется при копировании зон в верхнем левом углу экрана и может перетаскиваться в позицию вставки группы зон.

Размеры и позиция фокусной зоны на странице устанавливаются вводом числовых значений в со-

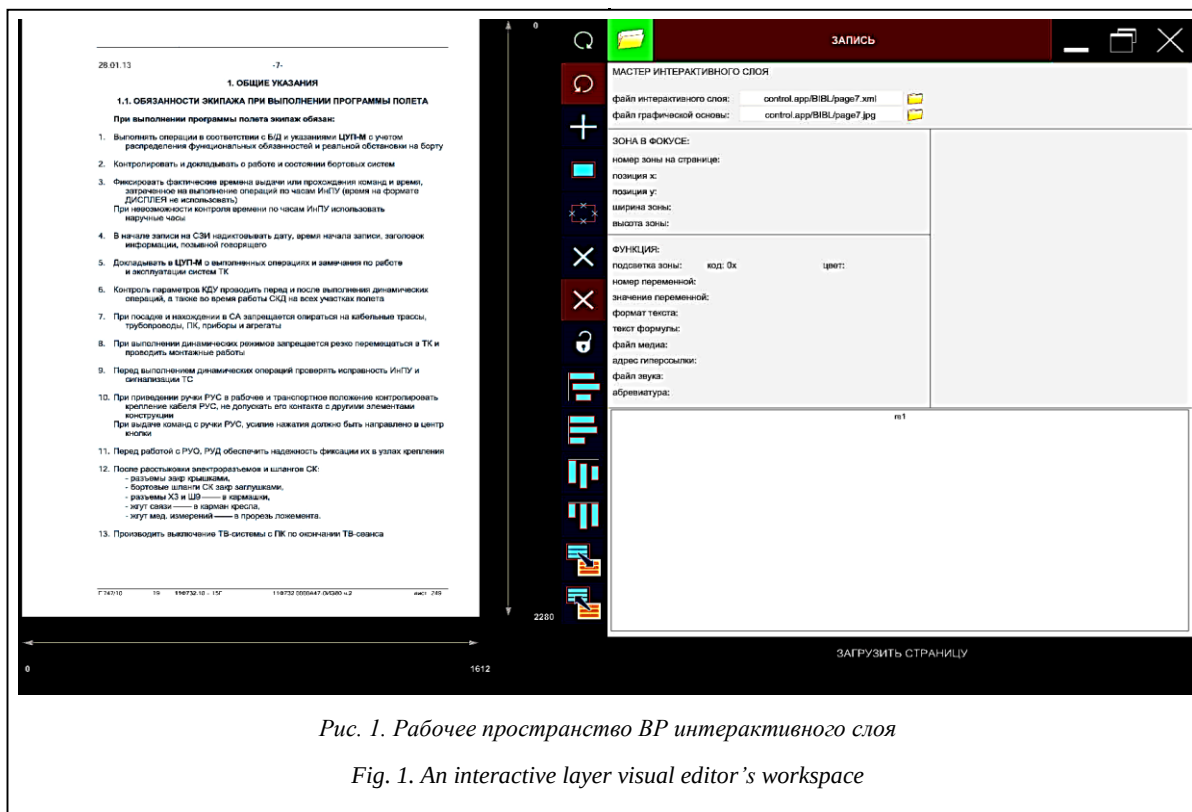


Рис. 1. Рабочее пространство ВР интерактивного слоя

Fig. 1. An interactive layer visual editor's workspace

ответствующие строки Мастера, захватом и перетаскиванием или перемещением движков на полях страницы.

Выделить зоны для редактирования можно нажатием на одиночную зону (она обводится красной рамкой) или растягиванием групповой, синей, рамки. Выделенными становятся только те зоны, которые полностью попадают в рамку.

При фокусировании единственной зоны в строках Мастера появляются ее параметры: номер зоны на странице, текущие позиции по горизонтали и вертикали, размеры по ширине и высоте.

Ниже в Мастере выводятся функциональные характеристики фокусной зоны: link – гиперссылка для перехода на заданную позицию в массиве ЭБД, abbrev (аббревиатура) – расшифровка принятых в системе сокращений, media (мультимедиа) – привязка к действию видео, анимации, графики, light – подсветка фокусного пункта (действия) при навигации, math – зона для вывода вычисляемого математического выражения, variable – зона для ввода переменных вычисляемого выражения, formular – строка для рукописных отметок в документе.

После создания на странице всех необходимых интерактивных зон и определения их функциональности производится запись файла управляющего слоя **page#.xml**. Итоговую композицию визуальной основы и управляющего слоя можно просмотреть и протестировать в редакторском экземпляре МИП.

Совокупность внедряемых на страницу ЭБД интерактивных и мультимедийных элементов

обеспечивает оператору новые функциональные возможности.

В качестве примера приведем функцию отложенного действия.

Некоторые пункты бортовых инструкций предусматривают некоторую временную задержку исполнения. Для блокировки органов управления МИП на время предписанной паузы предусмотрен специальный таймер отложенных действий (рис. 2), вызываемый атрибутом *time_out*="*<сек.>*" в тегах подсветки данного действия слоя **page#.xml**.

Таймер имеет круговую, цифровую и линейную индикации отсчета, кнопку экстренного сброса паузы. Линейная индикация дублируется на поле строки комментариев.

Пока идет отсчет паузы, все пространство экрана, за исключением кнопки экстренного сброса, скрыто затемнением, блокирующим доступ к органам управления МИП.

Завершение отсчета сопровождается звуковым оповещением. Интерактивная среда МИП разблокируется, оператор приступает к исполнению данного пункта ЭБД, его действия протоколируются.

Преобразование потоков обновляемых данных

Следует выделить три этапа переработки данных в процессе работы с ЭБД.

1. Этап ЗЕМЛЯ – все, что можно заранее сделать на Земле в офисных условиях. Это относительно медленные операции в произвольном вре-

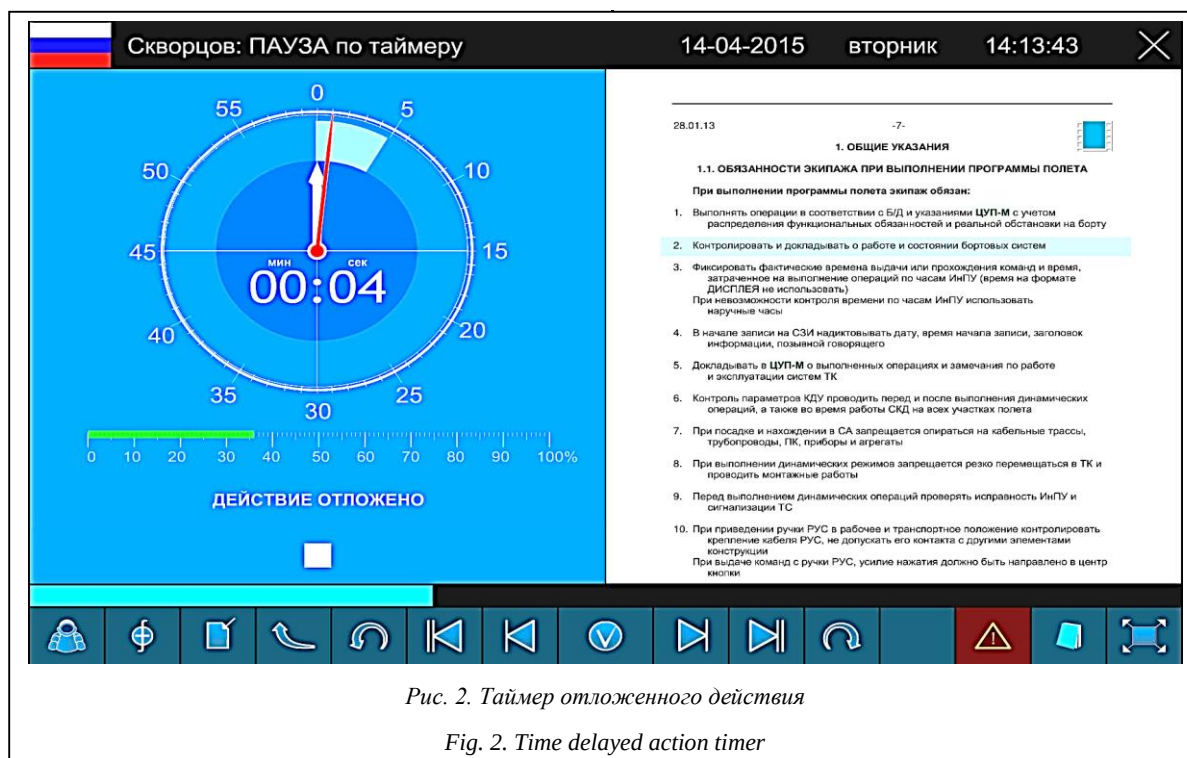


Рис. 2. Таймер отложенного действия

Fig. 2. Time delayed action timer

мени с привлечением относительно больших вычислительных ресурсов и квалифицированного персонала [9, 10].

2. Этап БОРТ – все, что можно сделать на борту при обновлении контента. Это также относительно медленные операции в произвольном времени с привлечением ограниченных ресурсов бортового сервера и без участия экипажа.

3. Этап МИП – все, что необходимо и достаточно для работы интерфейса мобильного терминала. Это быстрый доступ к разделам ЭБД, быстрое преобразование данных, высококачественное воспроизведение звукового сопровождения и растринированных изображений в темпе видео на разнородных вычислительных устройствах; экипаж не принимает участия в технологическом процессе, ограничиваясь только пользовательскими действиями; на этом этапе высока вероятность неконтролируемых аппаратных конфликтов и сбоев в работе.

Очевидно, что критическим аспектом приведенной классификации является соотношение операционной нагрузки и локальных вычислительных возможностей. Чем неопределеннее, ограниченнее, разнороднее вычислительные средства, используемые по мере продвижения процесса с Земли на борт, тем выше и строже требования к реализации алгоритмов переработки информации. Отсюда следует решение максимально освободить этап МИП от подготовительных задач переработки данных и перенести их на этапы ЗЕМЛЯ и БОРТ. На рисунке 3 приведены две последовательности операций переработки данных: традиционная

(верхняя гистограмма) и оптимизированная (нижняя гистограмма).

Цепочка тяжелых преобразований XML-XSL-XSLT-JavaScript-HTML-StageWebView, использовавшаяся в предыдущих версиях МИС (верхняя часть рис. 3), заменяется более «легкой» последовательностью XML-PDF-JPG+XML_inter (схема XPJX), в которой участвует опубликованный в наземных условиях документ PDF, за счет чего гарантируется идентичность вида процедуры на экране бортового терминала и утвержденного наземного бумажного документа. Разработка *нетиповых инструкций* (НИ) на Земле завершается формированием файла разметки, по которой на борту производится реконструкция НИ. Такое разделение позволяет избежать доставки на борт «тяжелых» элементов ЭБД. При этом в бортовом терминале выполняется обработка только связки JPG+XML_inter, преобразование XML-PDF осуществляется на Земле, а преобразование PDF-JPG, как и в случае с НИ, реконструируется на бортовом сервере. Планшетный терминал освобождается от самой опасной операции – кроссплатформенной конверсии потока данных в растровую форму, которая переносится из терминала в наземный и бортовой серверы.

Реализация интерфейса на различных ОС

Ключевым требованием при анализе возможности реализации МИП на компьютерах с различными ОС следует считать единство программного

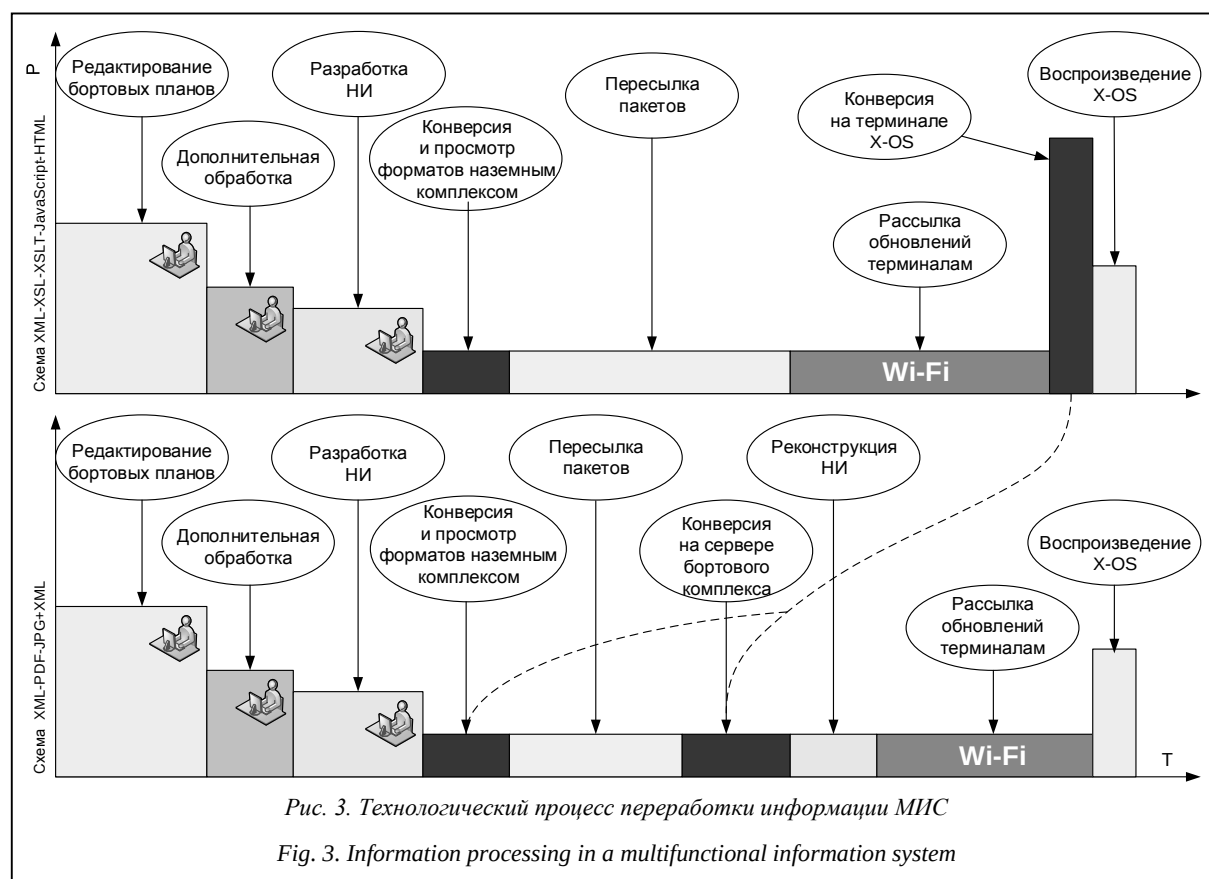


Рис. 3. Технологический процесс переработки информации МИС

Fig. 3. Information processing in a multifunctional information system

продукта, так как трудно поддерживать и адекватно модифицировать несколько версий системы, написанных под разные ОС. На опыте разработки нескольких прототипов в 2012–2015 годах и их исследованиях в реальных условиях космического полета установлено, что наиболее удобной и эффективной средой исполнения является AIR (Adobe Integrated Runtime), реализующая программы, написанные на языке сценариев ActionScript 3.0.

Приведенные выше концептуальные положения были реализованы при разработке базовой версии МИП и протестированы на настольных ПК и ноутбуках с ОС Windows, планшетах iPad (iOS) и Samsung (Android). Они показали высокую надежность, системную совместимость, гибкость конфигурирования, удобство эксплуатации и относительную простоту модернизации.

Созданный кластер новых систем интерактивного обеспечения деятельности экипажа существенно расширяет перспективы дальнейшего развития бортовой информационной среды пилотируемых космических объектов.

Литература

1. Операции и эксперименты на Международной космической станции 25 апреля 2013 года. URL: <http://www.federal-space.ru/main.php?id=2&nid=20060> (дата обращения: 15.05.2013).
2. Технические эксперименты и исследования на Российском сегменте МКС. Эксперимент «ВИРУ». URL: <http://www.energiya.ru/ru/iss/researches/techn/58.html> (дата обращения: 15.05.2013).

energiya.ru/ru/iss/researches/techn/58.html (дата обращения: 15.05.2013).

3. Кравченко С.И. Бортовой модуль информационной поддержки экипажа Международной космической станции // Программные продукты и системы. 2013. № 3 (103). С. 68–72.
4. Душенко А.Г., Арестов Д.С. Многофункциональная информационная система поддержки действий экипажа // Программные продукты и системы. 2013. № 3 (103). С. 54–61.
5. Осипов В.П., Сивакова Т.В., Судаков В.А., Трахтенгерц Э.А., Загребев Б.В. Методологические основы поддержки принятия решений при планировании научно-прикладных исследований и экспериментов на Международной космической станции (МКС) // Электротехнические и информационные комплексы и системы. 2013. № 3 (9). С. 81–88.
6. Жук Е.И., Гуторов А.М. Применение виртуальных руководств при подготовке к проведению научных исследований на пилотируемых орбитальных комплексах. Научные чтения памяти К.Э. Циолковского. Калуга, 2009. URL: <http://readings.gmik.ru/lecture/2009-primenenie-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemih-orbitalnih-kompleksah> (дата обращения: 15.05.2013).
7. Михеев А.М., Семочкина И.Ю. 3d-моделирование в интеллектуальной информационной системе поддержки научного эксперимента для систем мониторинга и контроля изделий ракетно-космической техники // Надежность и качество сложных систем. 2013. № 4. С. 28–32.
8. Забавникова Т.Ю. Элементы эргономики проблеме проектирования интерфейса // Вестн. Тамбовского ун-та. Сер.: Естествен. и технич. науки. 2009. № 1 (14). С. 227–228.
9. Мельникова Р.В. Проектирование пользовательского интерфейса // Восточно-Европейский журн. передовых технологий. 2010. № 8 (48). С. 18–20.
10. Быстров Д.А. Архитектуры презентационной системы образовательного насыщенного мультимедиа контента // Образовательные технологии и общество. 2007. № 4 (10). С. 329–338.

DOI: 10.15827/0236-235X.112.039-044

Received 09.09.15

ACTIVE INFORMATION SUPPORT OF CREW ACTIONS**Kravchenko S.I.**, Ph.D. (Engineering), Head of Sector, artstory@bk.ru;**Dushenko A.G.**, Ph.D. (Engineering), Head of Department;**Andreev D.A.**, Programmer Engineer, dimak134@yandex.ru

(Don Branch of the Space Simulator Center,

Platovsky Av. 101, Novocherkassk, 346400, Russian Federation);

Kotov A.G., Head of Department, alexander.kotov@sfoc.ru

(S.P. Korolev Rocket and Space Public Corporation Energia,

Lenina St. 4a, Korolev, Moscow Reg., 141070, Russian Federation)

Abstract. The article notes that the tasks of space ships and stations crews information support are continuously expanding due to onboard plans complexity, increasing scientific activities in an onboard schedule; increasing requirements to dispatch and clarity of the onboard documentation, importance of multimedia, additional background information and sound. The paper shows the results of experimental operation of a multifunctional information system prototype in the ISS. The experimental operation identified high-priority interactive objects and functions, which are necessary to ensure the effective work of the crew. They are: the possibility of sensor affection on the environmental model; changing the type, positions and internal parameters of the virtual objects; data input into interactive parts of the documents; marking the prescribed actions implementation; the prescribed time intervals execution; work with authorized bookmarks; questionnaire for an operator based on completed work session results; logging astronaut's behavior in the information environment and transferring protocols to the Earth. The authors formulate the problems of onboard information systems and their ground facilities. They also pay attention to increasing the capabilities of the interaction between an operator and information environment. The paper offers key solutions, explains new means of documents preparation. There are some examples of implementing active elements into environment. The paper considers the operator's situational behavior, work with virtual controls, logging the results and crew evaluation. It also describes the special editor of an e-document control layer. The authors offer solutions to overcome the uncertainty of choice of an operation system and personal tablets hardware platform.

Keywords: onboard documentation, interface module, multimedia, Russian segment of the ISS, interactive parts.

References

1. *Operatsii i eksperimenty na Mezhdunarodnoy kosmicheskoy stantsii 25 aprelya 2013 goda* [Operations and experiments at the ISS on April 25, 2013]. Available at: <http://www.federalspace.ru/main.php?id=2&nid=20060> (accessed May 15, 2013).
2. *Tekhnicheskie eksperimenty i issledovaniya na Rossiyskom segmente MKS. Eksperiment "VIRU"* [Technical experiments and researches on the Russian segment of ISS. "VIRU" experiment]. Available at: <http://www.energiya.ru/ru/iss/researches/techn/58.html> (accessed May 15, 2013).
3. Kravchenko S.I. The onboard module for information support of the international space station crew. *Programmnye produkty i sistemy* [Software & Systems], 2013, no. 3 (103), pp. 68–72 (in Russ.).
4. Dushenko A.G., Arestov D.S. Multifunctional crew activity support information system. *Programmnye produkty i sistemy* [Software & Systems], 2013, no. 3 (103), pp. 54–61 (in Russ.).
5. Osipov V.P., Sivakova T.V., Sudakov V.A., Trakhtengerts E.A., Zagreev B.V. Methodological base of decision-making support when planning scientific and applied research and experiments at the International Space Station (ISS). *Elektrotekhnicheskie i informatsionnye komplekсы i sistemy* [Electrical and data processing facilities and systems], 2013, no. 3 (9), pp. 81–88 (in Russ.).
6. Zhuk E.I., Gutorov A.M. Using virtual guides to prepare for scientific researches on piloted orbital segments. *Nauchnye chteniya pamyati K.E. Tsiolkovskogo* [Scientific readings in memory of K.E. Tsiolkovsky]. Kaluga, 2009. Available at: <http://readings.gmik.ru/lecture/2009-primeneniye-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemih-orbitalnih-kompleksah> (accessed May 15, 2013).
7. Mikheev A.M., Syomochkina I.Yu. 3d-modeling in the science experiment support intelligent information system for rocket-and-space monitoring and control system. *Nadezhnost i kachestvo slozhnykh system* [Quality and Reliability of Complex Systems], 2013, no. 4, pp. 28–32 (in Russ.).
8. Zabavnikova T.Yu. Ergonomics elements for the interface design problem. *Vestnik Tambovskogo universiteta. Seriya: Estestvennye i tekhnicheskie nauki* [Tambov Univ. Reports. Series: Natural and Technical Sciences], 2009, no. 1 (14), pp. 227–228 (in Russ.).
9. Melnikova R.V. User interface design. *Vostochno-Evropeysky zhurnal peredovykh technology* [Eastern-European Journal of Enterprise Technologies], 2010, no. 8 (48), pp. 18–20 (in Russ.).
10. Bystrov D.A. Architectures of a learning multimedia content presentation system. *Obrazovatelnye tekhnologii i obshchestvo* [Educational Technology and Society], 2007, no. 4 (10), pp. 329–338 (in Russ.).

УДК 004.588, 004.823

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.045-048

**БОРТОВОЙ КОМПЛЕКС
МНОГОФУНКЦИОНАЛЬНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ
МЕЖДУНАРОДНОЙ КОСМИЧЕСКОЙ СТАНЦИИ
(РОССИЙСКИЙ СЕГМЕНТ)**

С.И. Кравченко, к.т.н., начальник сектора, artstory@bk.ru;

Д.А. Андреев, инженер-программист;

Д.В. Брунько, инженер

(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия);

Е.Б. Горбачев, инженер, evgeniy.gorbachev@sfo.ru

(Ракетно-космическая корпорация «Энергия» им. С.П. Королева,
ул. Ленина, 4а, г. Королев, Московская обл., 141070, Россия)

В статье приведено описание нового бортового комплекса для представления членам экипажа Российского сегмента Международной космической станции электронной бортовой документации. Отмечается, что для обеспечения информационных систем пилотируемых космических кораблей и станций электронной бортовой документацией на Земле должна заранее выполняться разработка документов в специфических, совместимых с бортовым интерфейсом форматах. Выделяется наиболее ответственный этап технологического процесса подготовки – преобразование информации в бортовых условиях. Формулируются основные задачи бортового комплекса: прием и распаковка пакетов обновления, серверная конверсия документов в совместимый формат, сборка композиций по файлам наземной разметки, рассылка авторизованных обновлений на планшетные компьютеры экипажа, воспроизведение информации на планшетах и обеспечение индивидуальной поддержки космонавта в соответствии с его ролью, протоколирование работы персонала. Приводятся схема и состав утилит специального ПО бортового комплекса. Подробно описываются алгоритм работы элементов схемы и методология обновления ресурсов, состоящие в конверсии полученного описания страницы в графическую форму, позиционировании файлов на бортовом файл-сервере в индексированных папках, реконструкции документов при наличии в обновлении соответствующих файлов разметки, сборке документов в том виде, в каком они были сформированы на Земле. Особое внимание уделяется процессу обновления персональной базы документов на планшете пользователя, который может выполняться автоматически или по согласованию с оператором; при этом оператор инициирует обновление после получения извещения о готовности канала и нажимает на кнопку обновления функциональной панели персонального интерфейса. Подчеркивается, что все остальные сервисные операции с доставляемыми пакетами обновления выполняются автоматически, без участия экипажа.

Ключевые слова: бортовая документация, модуль интерфейса, мультимедиа, Российский сегмент МКС, бортовая информационная система.

Для обеспечения информационных систем пилотируемых космических кораблей и станций электронной бортовой документацией (ЭБД) [1] на Земле ведется разработка документов в специфических, совместимых с бортовым интерфейсом форматах [2]. Наиболее ответственным этапом технологического процесса подготовки ЭБД является преобразование информации в бортовых условиях [3].

После доставки пакета обновления на борту должны решаться следующие задачи:

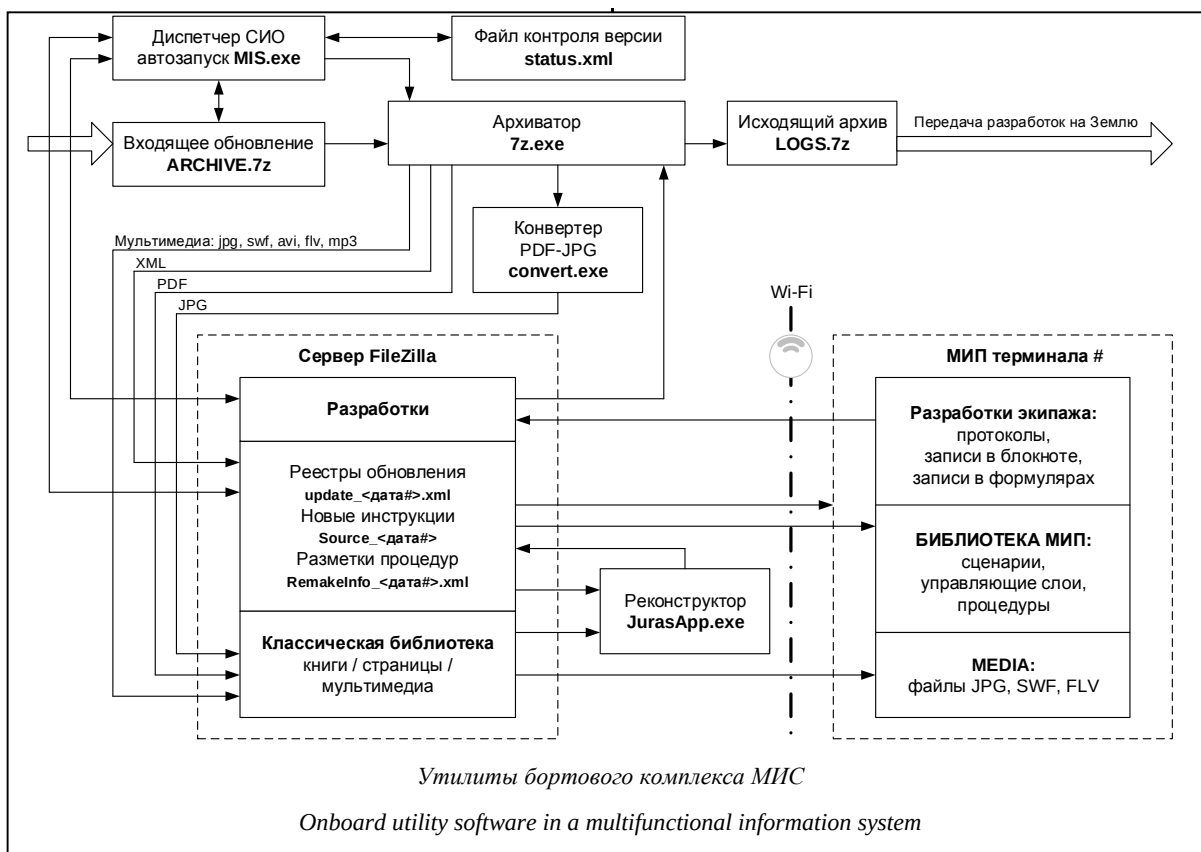
- прием и распаковка пакетов обновления ЭБД;
- серверная конверсия элементов ЭБД в совместимый формат;
- сборка композиций по файлам наземной разметки;
- рассылка авторизованных обновлений на планшетные компьютеры членов экипажа;
- воспроизведение ЭБД на планшетах, обеспечение полной информационной поддержки каждого космонавта в соответствии с его ролью [4];

– протоколирование работы персонала, передача пакетов бортовому серверу и далее на Землю для анализа и архивирования [5].

Для реализации такой последовательности на приемной стороне (на бортовом сервере RSK2) устанавливается комплекс специального ПО (см. рисунок).

В состав утилит бортового комплекса входят диспетчер системы информационного обеспечения (СИО) многофункциональной информационной системы (МИС) **MIS.exe**; архиватор **7z.exe**; конвертер форматов PDF – JPG **convert.exe**; файл-сервер **FileZilla**; реконструктор разметки процедур **JurasApp.exe**; терминальные модули интерфейса пользователя (МИП) на сервере и планшетах экипажа.

Диспетчер **MIS.exe** запускается на бортовом сервере в качестве объекта автозапуска. Он отслеживает доставку на борт обновления **archive.7z**. Версия входящего архива **archive.7z** сравнивается с мета-данными в файле **status.xml**, и, если входящий архив более поздний по времени доставки,



чем предыдущий, инициализируется процесс распаковки пуском программы **7z.exe** в режиме командной строки с аргументами:

<адрес архива 7z> -o<адрес папки назначения>.

Каждый полученный файл PDF-описания страницы конвертируется в графическую форму JPG модулем **convert.exe** в режиме командной строки с аргументами:

-density 300 <адрес pdf> -scale 1612 x 2280 -alpha remove <адрес jpg>.

Файлы JPG визуальной основы ЭБД позиционируются на сервере **FileZilla** в папку **Classic Library** (первичная загрузка) или в индексированные папки **Source_<дата#>** (для частичных досылок или нетиповых инструкций).

После этого в файле **status.xml** фиксируется время создания (версия) архива, из которого производилось текущее обновление.

Файлы реестров обновления **update_<дата#>.xml** и файлы разметок реконструкции **нетиповых инструкций (НИ) RemakeInfo_<дата#>** помещаются в корневую директорию сервера **FileZilla**. Там же находится серверное хранилище дополнительных иллюстраций и звукового сопровождения **Classic Library/Media**.

Следующим этапом обработки информации является реконструкция НИ при наличии в обновлении соответствующих файлов разметки **RemakeInfo_<дата#>**. По этим разметкам программа

JurasApp.exe выполняет «повторную сборку» документов в том виде, в каком они были сформированы на Земле. Источником материалов является **Classic Library**, результат реконструкции размещается в «перевалочной базе» **Source_<дата#>**.

Завершающий этап процесса – обновление персональной ЭБД на планшете пользователя по его инициативе. Оператор соглашается на обновление после получения на экране и в голосовой форме извещения о готовности канала. Он нажимает на кнопку обновления функциональной панели персонального МИП.

МИП обращается к соответствующему файлу **update_<дата#>.xml** и по содержащимся в нем атрибутам выполняет загрузку обновления в свою локальную библиотеку ЭБД **BIBL**. Процесс загрузки индицируется слайдером в нижней части экрана.

Предусмотрена также обратная передача данных по каналу борт–Земля. На Землю отправляются лог-файлы с протоколами поведения операторов и авторизованные разработки экипажа, например, результаты проектирования внутренних инструкций на автономном модуле планирования, записи в блокноте и т.п.

Передача осуществляется upload-командами ftp-протокола из МИП на файл-сервер **fileZilla**. Во временном хранилище данных сервера создается папка пользователя для аккумуляции лог-файлов, отправляемых на файл-сервер РС МКС. Формиру-

ется очередь пакетов данных на передачу в соответствии с содержимым папок МИП **Content/LOG<user>/log_<data>.xml**.

В заданной по компьютерному времени периодичности диспетчер **MIS.exe** пытается инициировать отправку данных файл-серверу РС МКС. Для этого проверяется папка **LOG**. Если папка не пуста, программа осуществляет ее упаковку в архив **LOGS.7z** пуском программы **7z.exe** в режиме командной строки с аргументами:

a <адрес архива 7z> <адрес папки LOG >.

Затем осуществляется перенос содержимого временного хранилища лог-файлов **LOG** в постоянное хранилище **Output Information**.

После успешной передачи лог-файлы из временного хранилища удаляются.

Исходящий архив **LOGS.7z** отправляется на Землю по расписанию работы канала связи борт-Земля средствами бортовой связи.

Таким образом, все сервисные операции с ЭБД выполняются автоматически, без участия экипажа.

Описанный бортовой комплекс МИС обеспечивает существенные преимущества по сравнению с бумажной документацией и предлагавшейся ранее в работе [4] технологией полной переработки данных в компьютерном терминале на рабочем месте космонавта.

Наиболее затратные и ненадежные вычислительные процедуры перенесены из бортовых алгоритмов реального времени на этап наземной офисной работы. Доставляемый на борт исходный массив обновляемых данных представляет собой полуфабрикат, готовый к быстрой переработке.

Значительная часть операций переработки осуществляется на бортовом сервере с единственной операционной системой (Windows). Роль персонального, как правило, планшетного, терминала сводится к сборке электронного документа из мультимедийных и интерактивных составляющих.

Существенно повышается надежность взаимосвязей между элементами страницы бортовой документации на всех этапах ее формирования и преобразования.

Появляется возможность реализации терминальных МИП на различных операционных систе-

мах (Windows, Android, iOS) с их минимальной взаимной адаптацией.

Впервые обеспечивается гибкая дисциплина контроля актуальности и ежедневного персонализированного обновления информации на личных планшетах членов экипажа в соответствии с их ролью и компетенцией [6].

Кроме того, значительно ускоряется реакция системы, что особенно важно при принятии решений в нештатных ситуациях [7], обеспечивается непрерывное протоколирование действий экипажа в информационном пространстве [8], а также реализуется комплекс мероприятий по ситуационному анализу и автоматизированной объективной оценке деятельности членов экипажа [9, 10] по итогам миссии.

Литература

1. Операции и эксперименты на Международной космической станции 25 апреля 2013 года. URL: <http://www.federal-space.ru/main.php?id=2&nid=20060> (дата обращения: 15.05.2015).
2. Технические эксперименты и исследования на Российском сегменте МКС. Эксперимент «ВИРУ». URL: <http://www.energiya.ru/ru/iss/researches/techn/58.html> (дата обращения: 15.05.2015).
3. Кравченко С.И. Бортовой модуль информационной поддержки экипажа Международной космической станции // Программные продукты и системы. 2013. № 3 (103). С. 68–72.
4. Душенко А.Г., Арестов Д.С. Многофункциональная информационная система поддержки действий экипажа // Программные продукты и системы. 2013. № 3 (103). С. 54–61.
5. Жук Е.И., Гуторов А.М. Применение виртуальных руководств при подготовке к проведению научных исследований на пилотируемых орбитальных комплексах: Науч. чтения памяти К.Э. Циолковского, Калуга, 2009. URL: <http://readings.gmik.ru/lecture/2009-primeneniye-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemykh-orbitalnih-kompleksah> (дата обращения: 15.05.2015).
6. Милицин А.В., Самсонов В.Н., Ходак В.А. и др. Отображение информации в Центре управления космическими полетами. М.: Радио и связь, 1982. 192 с.
7. Соловьев В.А., Лысенко Л.Н., Любинский В.Е. Управление космическими полетами. Ч. 2. Изд-во МГТУ им. Н.Э. Баумана, 2010. 205 с.
8. Беляев М.Ю. Научные эксперименты на космических кораблях и орбитальных станциях. М.: Машиностроение, 1984. 264 с.
9. Девятков В.В. Системы искусственного интеллекта. М.: Изд-во МГТУ им. Н.Э. Баумана, 2001. 352 с.
10. Любинский В.Е., Станиловская В.И. Планирование полета длительно функционирующих пилотируемых космических аппаратов // Космонавтика и ракетостроение. 2004. № 4 (37). С. 98–104.

DOI: 10.15827/0236-235X.112.045-048

Received 09.09.15

THE ONBOARD MULTIFUNCTIONAL INFORMATION COMPLEX OF THE INTERNATIONAL SPACE STATION (RUSSIAN SECTION)

Kravchenko S.I., Ph.D. (Engineering), Head of Sector, artstory@bk.ru;

Andreev D.A., Engineer-Programmer;

Brunko D.V., Engineer

(Don Branch of the Space Simulator Center,

Platovsky Av. 101, Novocherkassk, 346400, Russian Federation);

Gorbachev E.B., Engineer, evgeniy.gorbachev@sfo.ru

(S.P. Korolev Rocket and Space Public Corporation Energia,

Lenina St. 4a, Korolev, Moscow Reg., 141070, Russian Federation)

Abstract. The paper describes the new onboard set to provide the Russian ISS Segment crew with electronic documentation. It is noted that to support manned space ships and stations information systems with onboard electronic documentation,

the documents should be carried out in the formats compatible with onboard interface. The paper highlights the most important stage of the process of preparation. It is the onboard data conversion. The main tasks of onboard complex are reception and unpacking service packs, server conversion of documents in compatible format, compositions assembly by land marking files, mailing authorized updates to the crew's Tablet PCs, information reproduction on the plates and providing individual support of an astronaut according to his role, staff logging. The paper gives the scheme and the composition of the board complex special software utilities. It also describes in detail the algorithm of the circuit elements and the methodology of resources update, which mean converting received page description in a graphic form, files positioning in the onboard file server, documents updating when there are appropriate markup files, documents collection in the form in they were formed on the Earth. Particular attention is paid to the process of updating the personal documents database on the user's tablet that can be run automatically or by agreement with the operator, who initiates the update after receiving the notice and clicks the refresh button on personal interface panel. It is emphasized that all other service operations with delivered updates are performed automatically without crew.

Keywords: onboard documentation, interface module, multimedia, Russian segment of the ISS, onboard information system.

References

1. *Operatsii i eksperimenty na Mezhdunarodnoy kosmicheskoy stantsii 25 aprelya 2013 goda* [Operations and experiments at the ISS on April 25, 2013]. Available at: <http://www.federalspace.ru/main.php?id=2&nid=20060> (accessed May 15, 2015).
2. *Technical experiments and researches on the Russian segment of ISS. "VIRU" experiment*. Available at: <http://www.energia.ru/ru/iss/researches/techn/58.html> (accessed May 15, 2015).
3. Kravchenko S.I. The onboard module for information support of the international space station crew. *Programmnye produkty i sistemy* [Software & Systems], 2013, no. 3 (103), pp. 68–72 (in Russ.).
4. Dushenko A.G., Arestov D.S. Multifunctional crew activity support information system. *Programmnye produkty i sistemy* [Software & Systems], 2013, no. 3 (103), pp. 54–61 (in Russ.).
5. Zhuk E.I., Gutorov A.M. Using of the virtual managements for preparation to scientific researches on piloted orbital complexes. *Nauchnye chteniya pamyati K.E. Tsiolkovskogo* [Scientific readings in memory of K.E. Tsiolkovsky]. Kaluga, 2009. Available at: <http://readings.gmik.ru/lecture/2009-primeneniye-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemyh-orbitalnih-kompleksah> (accessed May 15, 2015).
6. Militsin A.V., Samsonov V.N., Khodak V.A. *Otobrazhenie informatsii v Tsentre upravleniya kosmicheskimi poletami* [Display information in the Mission Control Center]. Moscow, Radio i svyaz Publ., 1982, 192 p.
7. Solovyov V.A., Lysenko L.N., Lyubinsky V.E. *Upravlenie kosmicheskimi poletami* [Mission Control]. Part 2. N.E. Bauman MSTU Publ., 2010, 205 p.
8. Belyaev M.Yu. *Nauchnye eksperimenty na kosmicheskikh korablyakh i orbitalnykh stantsiyakh* [Scientific Experiments on Spacecrafts and Orbital Stations]. Moscow, Mashinostroyeniye Publ., 1984, 264 p.
9. Devyatkov V.V. *Sistemy iskusstvennogo intellekta* [Artificial Intelligence Systems]. Moscow, N.E. Bauman MSTU Publ., 2001, 352 p.
10. Lubinsky V.E., Stanilovskaya V.I. Flight Planning for Long Operating Manned Spacecrafts. *Kosmonavtika i raketostroyeniye* [Cosmonautics and Rocket Engineering]. 2004, no. 4 (37), pp. 98–104.

Реклама

Центр тренажеростроения и подготовки персонала

предлагает свои услуги по созданию молодежных
образовательно-познавательных-развлекательных Космоцентров под ключ
в любом городе России.

Подробности на сайте
www.simct.ru

УДК 004.588, 004.823

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.049-054

НАЗЕМНАЯ ПОДСИСТЕМА ПОДГОТОВКИ ЭЛЕКТРОННОЙ БОРТОВОЙ ДОКУМЕНТАЦИИ

С.И. Кравченко, к.т.н., начальник сектора, artstory@bk.ru;

Д.А. Андреев, инженер-программист;

А.А. Долгов, инженер-программист

*(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия);*

А.В. Колесников, заместитель начальника отдела, alexander.kolesnikov@sfo.ru

*(Ракетно-космическая корпорация «Энергия» им. С.П. Королева,
ул. Ленина, 4а, г. Королев, Московская обл., 141070, Россия)*

Описан технологический процесс переработки информации при подготовке электронной бортовой документации для многофункциональной информационной системы поддержки деятельности экипажей транспортных космических кораблей и орбитальных станций. Подчеркивается необходимость генерации электронных документов на этапе бортового обновления контента из совместимого доставляемого архива. Отмечается концептуальная особенность наземной подсистемы, заключающаяся в разделении элементов электронного документа на две категории: консервативную визуальную основу и интерактивный слой, состоящий из виртуальных органов управления разделами документа и внутренними приложениями. Формулируются преимущества раздельной обработки слоев документа, состоящие в наземной разметке инструкции и доставке на борт только файла разметки, по которому на этапе обновления автоматически реконструируется разработанная на Земле инструкция; таким образом достигается идентичность наземной и бортовой версий, существенно снижается нагрузка на канал передачи данных Земля–борт. В статье также приводится структура средств наземного комплекса, раскрывается последовательность выполняемых операций; дается детальное описание взаимодействия участвующих в процессе утилит и основных разделов общей методологии: архивирование разработок в классической библиотеке бортовой документации, генерация визуальной основы, редактирование управляющего слоя, формирование новых документов из массива страниц классической библиотеки. Особое внимание уделяется специальным инструментам – редактору-компоновщику разметок и формирователю пакетов обновления. На основании экспериментов делается заключение о готовности разработанного комплекса наземных средств подготовки электронной бортовой документации к полноценному обеспечению формирования пакетов обновления для многофункциональной информационной системы поддержки деятельности экипажей космических кораблей и станций.

Ключевые слова: бортовая документация, модуль интерфейса, мультимедиа, Российский сегмент МКС, информационная система.

Технологический процесс переработки информации для космических интерфейсов [1] базируется на следующих основных положениях.

Вводится строгое разделение элементов *электронной бортовой документации* (ЭБД) на две категории:

- визуальная основа – «внешний вид» процедуры (страницы), соответствующий стандарту ODF, это то, что можно напечатать на бумаге [2];
- интерактивный (управляющий) слой, состоящий из виртуальных органов управления ЭБД и приложений (кнопки, подсветка пунктов, отметки об исполнении, зоны для генерации заполняемых формуляров, указатели расшифровок аббревиатур, гиперссылки, значки для запуска внешних мультимедийных приложений, дикторские комментарии к пунктам процедуры, записи в авторизованном блокноте и др.) [3].

Процесс генерации визуальной основы выполняется на этапе бортового обновления контента *многофункциональной информационной системы* (МИС). *Модуль интерфейса пользователя* (МИП) получает пакет готовых решений. Отпадает необходимость реализации в реальном времени на разнородных (часто низкопроизводительных) ап-

паратных средствах многоступенчатой сборки страницы. Повышается надежность работы терминала независимо от его ОС, снижается время доступа к процедурам, обеспечивается полная адекватность ЭБД исходному (утвержденному) документу [4].

Процесс формирования нетиповых (нестандартных) инструкций разделяется на два этапа. Сначала инструкция полностью создается в наземных условиях; при этом генерируется файл разметки с указанием составных частей новой композиции и их локализации в общей ЭБД; затем на борт доставляется только файл разметки, по которому на этапе обновления автоматически реконструируется разработанная на Земле инструкция. Достигается идентичность наземной и бортовой версий нетиповой инструкции, существенно снижается нагрузка на канал передачи данных Земля–борт [5].

Приведенные концептуальные положения реализованы и протестированы при разработке нескольких версий МИП [2] для настольных компьютеров под ОС Windows, планшетов iPad (iOS), Samsung (Android). Прототипы показали высокую надежность, системную совместимость, гибкость

конфигурирования, удобство эксплуатации и относительную простоту модернизации.

Структура наземного комплекса (НК) МИС приведена на рисунке 1.

Утилиты НК МИС реализуют технологическую последовательность подготовки ЭБД:

- подготовка исходного (бумажного) документа БД;
- студийная подготовка сюжетов мультимедиа;
- архивирование разработок в классической библиотеке БД;
- разработка сценария;
- генерация визуальной основы;
- создание и визуальное редактирование управляющего слоя;
- формирование процедур из массива страниц;
- контроль результата на наземном интерфейсе;
- формирование пакета и реестра обновлений;
- подготовка отчетов по возвращаемым с орбиты протоколам.

Средства разработки и редактирования книг БД представляют собой стандартные редакторы класса WISIWIG (What I see is what I get – вижу то, что получаю), например MS WORD, CorelDraw и т.п. [6]; или редакторы скриптового класса, например Arbortext, Dreamviewer и другие XML- и

HTML-редакторы [7]. Первые предпочтительнее, так как позволяют оператору работать непосредственно в визуальной среде и обеспечивают однозначность результата, воспроизводимого различными ОС и техническими платформами. Результатом работы средств является экспортированный комплект книг БД в формате PDF, утверждаемых контролирующими службами и архивируемых в наземной библиотеке, например Windchill.

Одновременно в НК разрабатывается мультимедийное иллюстративное обеспечение (видео, анимация, полноэкранная графика, звуковое сопровождение) [8], которое не входит в состав классической БД и является расширяющим функциональным дополнением ЭБД в среде МИП.

Структура библиотеки БД форматируется Редактором сценариев в xml-описание, которое отражает иерархическую вложенность книг/разделов/процедур. Сценарии типа description.xml описывают как полную ЭБД, так и нетиповые инструкции, базирующиеся на композициях классических процедур или разрабатываемые дополнительно.

Сценарий для ЭБД разрабатывается на стандартных офисных средствах.

На следующем этапе книги автоматически конвертируются в отдельные страницы PDF и их визуальные копии JPG (визуальная основа). Конверсия осуществляется утилитой **PDFDivider.exe**.

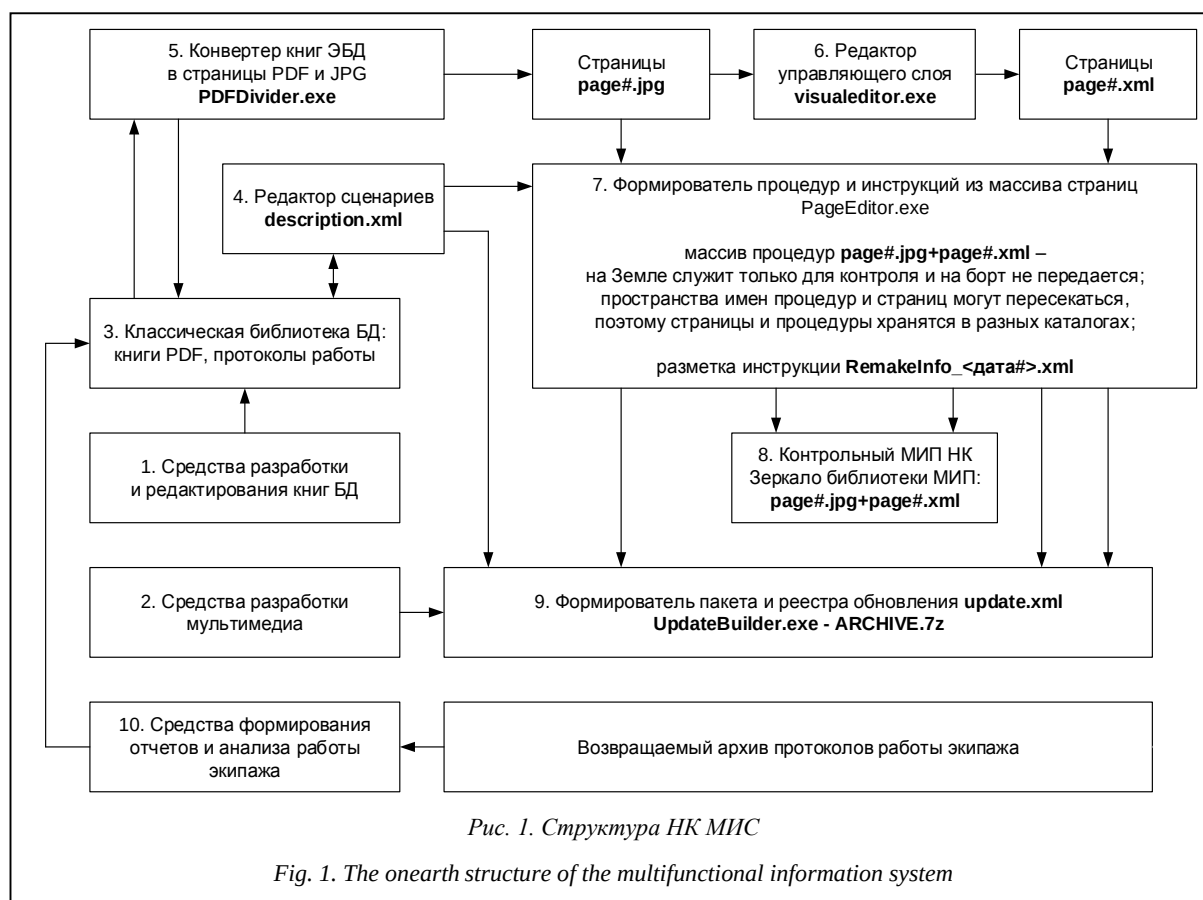


Рис. 1. Структура НК МИС

Fig. 1. The onearth structure of the multifunctional information system

Файлы основы передаются *визуальному редактору* (BP) управляющего слоя **visualeditor.exe**, в среде которого выполняется формирование интерактивных экранных зон с различной реакцией. Итогом работы на BP являются страницы ЭБД **page#.xml**.

Пары страниц «основа/интерактив» передаются непосредственно на упаковку и отправку на борт, а также используются для формирования разметки процедур из нескольких страниц классической ЭБД или их фрагментов. Этот этап реализуется утилитой **WindowsEditor.exe** и применяется также для разметки нетиповых инструкций.

Функциональность и связность слоистой композиции проверяются НК МИП.

На упаковку передаются

- сценарии библиотеки, полетных операций, нетиповых инструкций;
- стандартные страницы книг PDF;
- файлы управляющего интерактивного слоя;
- файлы разметки процедур и нетиповых инструкций.

Формирователь пакета и реестра обновлений **UpdateBuilder.exe** готовит к отправке на борт исходящий архив **ARCHIVE.7z**. На этом первичная работа НК завершается.

В дальнейшем на Землю возвращаются протоколы деятельности экипажа, которые обрабатываются средством формирования отчетов. Отчеты анализируются, оцениваются и размещаются в архиве корпорации [9].

Генерация визуальной основы

Визуальная основа представляет собой графическую копию (JPG) векторной страницы (PDF). Это будущая подложка экранной формы.

Обработке подвергается целая книга БД. Она разбивается на отдельные PDF-страницы, которые в дальнейшем доставляются на борт и включаются в основную, классическую библиотеку ЭБД. По ним программой **PDFDivider.exe** осуществляется генерация визуальных основ.

Создаваемые при генерации графические файлы используются только на Земле. По ним ведутся разработка управляющих слоев, разметка нетиповых процедур, сшивка страниц в типовые процедуры и т.п. На борту JPG-страницы генерируются повторно аналогичным, но уже полностью автоматическим инструментом, и не из книги, а из массива страниц PDF. Таким образом, JPG отдельно на борт не доставляются и не нагружают канал связи.

Разработка сценария

Распределение книг в разделах библиотеки ЭБД, структура самих книг до уровня процедур и действий должны быть представлены так называемым

сценарием, который отражает их иерархическую вложенность [10].

Сценарий **description.xml** может описывать или полную ЭБД, или нетиповые инструкции, базирующиеся на композициях классических процедур, или временные полетные операции.

Объем текста сценария относительно невелик и может разрабатываться на любом офисном редакторе xml.

Создание и визуальное редактирование управляющего слоя

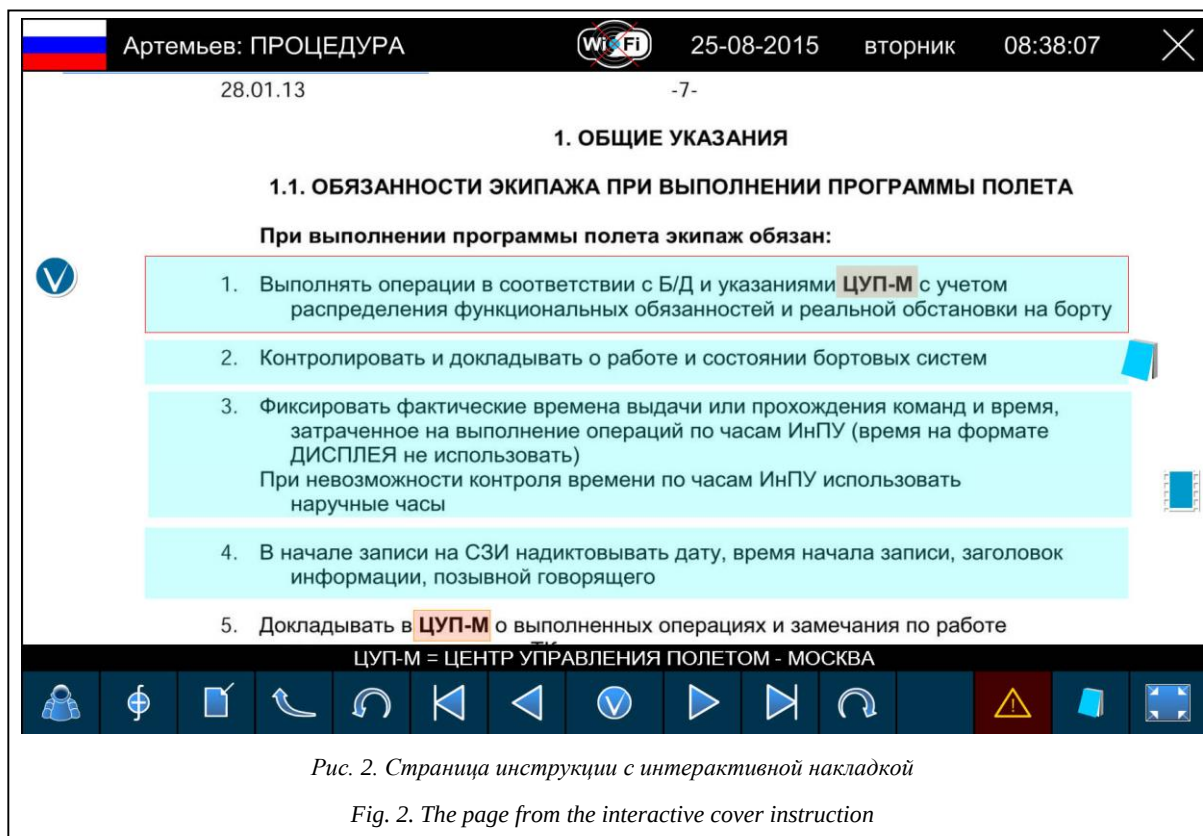
Файлы основы **page#.jpg** последовательно загружаются в BP управляющего слоя **visualeditor.exe**, в среде которого выполняется формирование интерактивных экранных зон с различной реакцией. Итогом работы BP являются страницы-накладки **page#.xml**. Накладки и файлы визуальной основы образуют пары, обрабатываемые синхронно. Впоследствии результат обработки (нарезки фрагментов, компоновки нетиповых инструкций и др.) воспроизводится МИП и обеспечивает заданную функциональность. На рисунке 2 приведен фрагмент инструкции на экране МИП с тремя типами внедренных активных зон управляющего слоя: сенсорные зоны пунктов инструкции (голубая подсветка, фокусный пункт 1 выделен рамкой), аббревиатура («ЦУП-М» – при наведении курсора в нижней строке подсказки выводится ее расшифровка), медиа-иллюстрация к пункту 3 (значок «кадр киноленты» – нажатие вызывает полноэкранный проигрыватель, воспроизводящий видео). Уже в среде МИП сделаны отметки о выполнении пункта 1 (птичка на левом поле) и о наличии записи в блокноте к пункту 2.

Формирование процедур из массива страниц

В ЭБД МКС, в отличие от ЭБД транспортного корабля «Союз», информационной единицей является не страница книги, а процедура. Чтобы из массива страниц классической библиотеки сформировать массив процедур, нужно предварительно создать файл разметки **RemakeInfo.xml**, для чего выполнить следующие операции:

- перечислить страницы, входящие в состав каждой процедуры (выполняет оператор);
- удалить на них колонтитулы и пустоты (выполняет автомат);
- сшить ленты процедур визуально (выполняет оператор) или формально – по списку входимости (выполняет автомат);
- параллельно выполнить аналогичные операции над управляющими слоями (выполняет автомат).

Для этих целей создан модуль-компоновщик процедур и нетиповых инструкций **PageEditor.exe**.



Библиотека представляет собой множество книг, книги отображаются в списке «Книга», страницы выбранной книги отображаются в списке «Страница».

Наполнение новой страницы контентом осуществляется визуальным копированием, вставкой, перемещением областей классических страниц. Действия, доступные на данной странице, представлены в меню «Редактирование».

Фрагменты процедуры на новой странице можно увидеть в окне «Фрагменты страницы». Выбранный на новой странице фрагмент можно перемещать перетаскиванием.

Пункты меню «Файл/Сохранить» и «Файл/Загрузить» позволяют сохранить или загрузить текущие наброски в формате данных компоновщика (.cdat).

Выбор пункта меню «Файл/Экспорт» запускает автоматическое построение инструкции и ее xml-описания, а также формирование соответствующего файла RemakeInfo.xml в выбранной пользователем папке.

Более формализованным и автоматизированным методом компоновки является композиция процедуры по списку страниц и перечисленных для них пунктов.

В редакторе запускается пункт меню «Редактирование/Сформировать из xml». Открывается форма для создания новой страницы. Ее интерфейс схож с интерфейсом визуальной компоновки. Книги отображаются в списке «Книги».

Страницы выбранной книги отображаются во втором столбце и выстраиваются в древовидную структуру.

Для выбранной страницы отображается список «Зона» с перечислением созданных для этой страницы навигационных зон.

Для визуализации действий на форме имеются два окна. В левом отображается текущая выбранная страница и подсвечиваются выбранные зоны. В правом выводится макет формируемой страницы (процедуры) (рис. 3).

При нажатии кнопки «Сформировать» форма с композицией закрывается, а сформированная процедура отображается в списке новых страниц на главной форме.

Формирование пакета и реестра обновления

Для доставки обновления на борт нужно сформировать пакет файлов, вложить в него реестр обновления и файлы разметок, упаковать все эти элементы в архив и передать доставляющей службе.

Реестр обновления **update.xml** – это файл, в котором хранятся инструкции, используемые на борту модулем МИП для обновления его библиотеки и сервером СИО МИС (MIS.exe) для обновления классической библиотеки МИС. Реестр **update.xml** упаковывается в единый архив с перечисленными в нем элементами контента и/или файлом разметки **RemakeInfo.xml**.

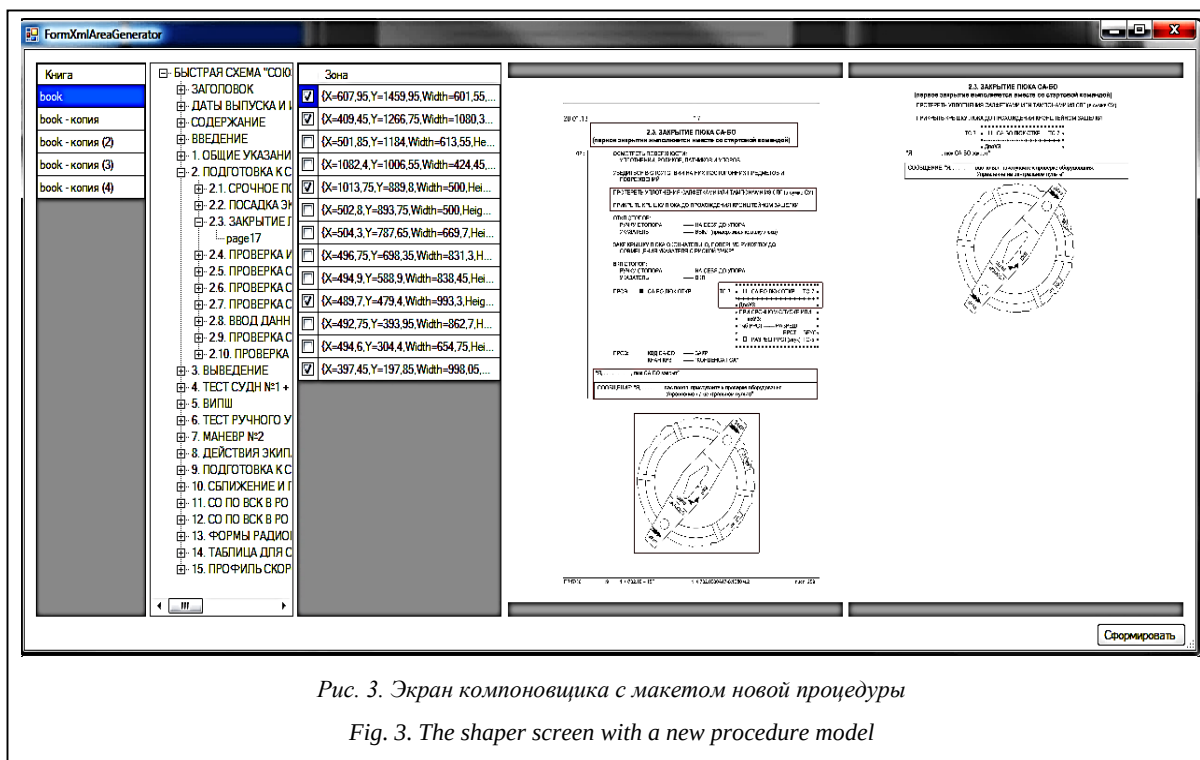


Рис. 3. Экран компоновщика с макетом новой процедуры

Fig. 3. The shaper screen with a new procedure model

Создан автоматизированный инструмент **UpdateBuilder.exe**, в среде которого выполняются следующие операции:

- создание нового документа **update.xml**;
- загрузка существующего документа;
- сохранение нового или отредактированного реестра обновлений;
- автоматическое архивирование пакета обновления для отправки на борт; в архив включаются сформированный реестр обновлений, файлы-источники, привязанные к элементам пакета обновлений, и опционально – файл разметки нетиповых процедур.

Разработанный комплекс наземных средств подготовки ЭБД обеспечивает полноценное формирование пакетов обновления для МИС поддержки деятельности экипажей космических кораблей и станций.

Литература

1. Операции и эксперименты на Международной космической станции 25 апреля 2013 года. URL: <http://www.federal-space.ru/main.php?id=2&nid=20060> (дата обращения: 15.05.2015).
2. Кравченко С.И. Бортовой модуль информационной поддержки экипажа Международной космической станции // Программные продукты и системы. 2013. № 3 (103). С. 68–72.

3. Душенко А.Г., Арестов Д.С. Многофункциональная информационная система поддержки действий экипажа // Программные продукты и системы. 2013. № 3 (103). С. 54–61.

4. Осипов В.П., Сивакова Т.В., Судаков В.А., Трахтенгерц Э.А., Загребев Б.В. Методологические основы поддержки принятия решений при планировании научно-прикладных исследований и экспериментов на Международной космической станции (МКС) // Электротехнические и информационные комплексы и системы. 2013. № 3 (9). С. 81–88.

5. Жук Е.И., Гуторов А.М. Применение виртуальных руководств при подготовке к проведению научных исследований на пилотируемых орбитальных комплексах. Науч. чтения памяти К.Э. Циолковского, Калуга, 2009. URL: <http://readings.gmik.ru/lecture/2009-primeneniye-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemih-orbitalnih-kompleksah> (дата обращения: 15.05.2015).

6. Михеев А.М., Семочкина И.Ю. 3D-моделирование в интеллектуальной информационной системе поддержки научного эксперимента для систем мониторинга и контроля изделий ракетно-космической техники // Надежность и качество сложных систем. 2013. № 4. С. 28–32.

7. Забавникова Т.Ю. Элементы эргономики проблеме проектирования интерфейса // Вестн. Тамбовского ун-та: Сер. Естественные и технические науки. 2009. № 1 (14). С. 227–228.

8. Елистратова Н.Н. Некоторые проблемы применения мультимедиа в системе высшего образования // Вестн. Рязанского гос. ун-та им. С.А. Есенина. 2010. № 26. С. 22–28.

9. Мельникова Р.В. Проектирование пользовательского интерфейса // Восточ.-Европ. журн. передовых технологий. 2010. № 8 (48). С. 18–20.

10. Быстров Д.А. Архитектуры презентационной системы образовательного насыщенного мультимедиа контента // Образовательные технологии и общество. 2007. № 4 (10). С. 329–338.

Kolesnikov A.V., Deputy Head of Department, alexander.kolesnikov@sfo.ru
(S.P. Korolev Rocket and Space Public Corporation Energia,
Lenina St. 4a, Korolev, Moscow Reg., 141070, Russian Federation)

Abstract. The paper describes information processing when preparing electronic onboard documents for multi-function information system to support a crew of a transport spacecraft and space stations. It is necessary to generate electronic documents from a delivered compatible file at the stage of the onboard content update. The conceptual feature of the ground subsystem is a separation of electronic document elements into two categories, which are: a conservative visual basis and an interactive layer that includes virtual controls for the sections of the document and internal applications. The paper considers the benefits of separate processing of document layers that consist in instruction ground marking and delivering on board only a cue file, which allows reconstructing an instruction developed on the Earth automatically. This makes the ground and onboard versions identical and significantly reduces the load on the data-transmission channel. The article also shows the ground-based complex structure, the sequence of the operations; a detailed description of the interaction between the utilities and major sections of the common methodology, such as: archiving developments in a classical library database, generating a visual base, editing a control layer, forming new documents from a classic library page array. The particular attention is paid to special tools that are: a link editor and the update package shaper. Based on experiment results, there is a conclusion about the readiness of the developed ground-based complex to ensure forming of update packages for the multifunctional information support system for a crew of space ships and stations.

Keywords: onboard documentation, interface module, multimedia, Russian segment of ISS, information system.

References

1. *Operatsii i eksperimenty na Mezhdunarodnoy kosmicheskoy stantsii 25 aprelya 2013 goda* [Operations and experiments at the ISS on April 25, 2013]. Available at: <http://www.federspace.ru/main.php?id=2&nid=20060> (accessed May 15, 2015).
2. Kravchenko S.I. The onboard module for information support of the international space station crew. *Programmnye produkty i sistemy* [Software & Systems], 2013, no. 3 (103), pp. 68–72 (in Russ.).
3. Dushenko A.G., Arestov D.S. Multifunctional crew activity support information system. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 3 (103), pp. 54–61 (in Russ.).
4. Osipov V.P., Sivakova T.V., Sudakov V.A., Trakhtengerts E.A., Zagreev B.V. Methodological base of decision-making support when planning scientific and applied research and experiments at the International Space Station (ISS). *Elektrotekhnicheskie i informatsionnye komplekсы i sistemy* [Electrical and data processing facilities and systems], 2013, no. 3 (9), pp. 81–88 (in Russ.).
5. Zhuk E.I., Gutorov A.M. Using of the virtual managements for preparation to scientific researches on piloted orbital complexes. *Nauchnye chteniya pamyati K.E. Tsiolkovskogo* [Scientific readings in memory of K.E. Tsiolkovsky]. Kaluga, 2009. Available at: <http://readings.gmik.ru/lecture/2009-primeneniye-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemih-orbitalnih-kompleksah> (accessed May 15, 2015).
6. Mikheev A.M., Syomochkina I.Yu. 3d-modeling in the science experiment support intelligent information system for rocket-and-space monitoring and control system. *Nadezhnost i kachestvo slozhnykh sistem* [Quality and Reliability of Complex Systems]. Penza, 2013, no. 4, pp. 28–32 (in Russ.).
7. Zabavnikova T.Yu. Ergonomics elements for the interface design problem. *Vestnik Tambovskogo universiteta. Seriya: Estestvennye i tekhnicheskie nauki* [Tambov Univ. Reports. Series: Natural and Technical Sciences], 2009, no. 1 (14), pp. 227–228 (in Russ.).
8. Elistratova N.N. Some problems and drawbacks of multimedia use in higher education system. *Vestn. Ryazanskogo gos. un-ta im. S.A. Esenina* [Bulletin of S.A. Esenin Ryazan State Univ.]. 2010, no. 1(26), pp. 22–28 (in Russ.).
9. Melnikova R.V. User interface design. *Vostochno-Evropeysky zhurnal peredovykh technology* [Eastern-European Journal of Enterprise Technologies], 2010, no. 8 (48), pp. 18–20 (in Russ.).
10. Bystrov D.A. Architectures of a learning multimedia content presentation system. *Obrazovatelnye tekhnologii i obshchestvo* [Educational Technology and Society], 2007, no. 4 (10), pp. 329–338 (in Russ.).

УДК 004.588, 004.823

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.055-057

ИНФОРМАЦИОННАЯ ПОДДЕРЖКА КОСМИЧЕСКИХ ЭКСПЕРИМЕНТОВ*В.В. Степанов, ведущий инженер;**Д.А. Андреев, инженер-программист; Е.А. Геркен, инженер**(Донской филиал Центра тренажеростроения,**просп. Платовский, 101, г. Новочеркасск, 346400, Россия, artstory@bk.ru);**С.С. Обыденков, ведущий инженер, gcn2@sfo.ru**(Ракетно-космическая корпорация «Энергия» им. С.П. Королева,**ул. Ленина, 4а, г. Королев, Московская обл., 141070, Россия)*

В статье приводятся результаты апробации комплекса виртуальных руководств космическими экспериментами на борту Российского сегмента Международной космической станции. На основе полученного опыта ставится задача перевода научной документации на планшетные компьютеры экипажа. Дается описание нового бортового модуля интерфейса с расширенными функциональными возможностями: версионностью сценария с несколькими уровнями сложности и переключением версий во время работы; расширенным аппаратом настройки начальных условий для более наглядного и точного воспроизведения планируемой работы; доступностью выбора последовательности действий и оперативного ввода начальных условий непосредственно на борту. При этом варьируются сценарии работы и цепочки предписанных действий, состав иллюстративного материала и используемого оборудования, виртуальные модели окружающей среды и показания приборов, значения даты и времени. Отмечается специфическая особенность информационной поддержки космических экспериментов, состоящая в необходимости параллельного сопровождения действий экипажа двумя сюжетными линиями – заранее подготовленной сценарной последовательностью, моделирующей виртуальную среду эксперимента, и произвольным доступом к классическому массиву бортовой документации; при этом ключевые точки виртуального процесса должны быть однозначно увязаны с пунктами документации. Описаны также системные решения и методология интерактивного управления информационным пространством. Уделяется внимание ситуационному поведению оператора, работе с виртуальными органами управления, протоколированию результатов и оценке деятельности экипажа. Подчеркивается, что работа космонавта полностью авторизована на уровне персональных сценариев, которые передаются ему в соответствии с индивидуальной ролью в коллективных экспериментах. Прогнозируются дальнейшее расширение функциональных возможностей программно-аппаратного комплекса информационной поддержки космических экспериментов и существенное повышение удобства, наглядности, оперативности действий операторов, получение высокой скорости доступа к материалам бортовой документации.

Ключевые слова: космический эксперимент, виртуальное руководство, мультимедиа, Российский сегмент МКС, моделирование, циклограмма, интерфейс.

Одной из главных миссий человека в космосе являются научные исследования. Для их проведения разрабатываются методические документы и комплекс средств информационной поддержки [1]. В настоящее время идет процесс переноса бортовых инструкций с традиционных бумажных носителей на компьютерную основу [2].

С 2011 года экипажами Российского сегмента Международной космической станции (МКС) проводится космический эксперимент ВИРУ [3]. Задача эксперимента – проверить на примере нескольких научных работ («Ураган», «Релаксация» и др.) возможность и эффективность использования для информационной поддержки специально разработанных виртуальных руководств научными космическими экспериментами.

По итогам использования комплекса ВИРУ космонавтами и наземным персоналом был сделан однозначный вывод о целесообразности использования на борту компьютерной (бесбумажной) бортовой документации [4]. В результате исследований создан базис для широкого внедрения на борту МКС виртуальных руководств. В то же время значительно расширились требования к функциональным возможностям *электронной бортовой документации* (ЭБД), в частности, поставлен вопрос об

использовании для ее обработки и воспроизведения планшетных компьютеров.

До настоящего времени экспертами не принято окончательное решение о выборе мобильной ОС, поэтому создаваемые планшетные прототипы имеют возможность работы на трех основных ОС: Windows, iOS и Android.

Разработана перспективная модификация комплекса ВИРУ в планшетной реализации, обеспечивающая все отработанные ранее функциональные возможности (рис. 1).

По рекомендациям экипажей в систему заложены дополнительные функции.

1. Обеспечена версионность сценария. Система настраивается на несколько степеней сложности представления информации с возможностью перехода от версии к версии во время работы.

2. Введен расширенный аппарат настройки начальных условий для более наглядного и точного воспроизведения планируемой работы. Обеспечен выбор того или иного варианта последовательности действий. Перед проведением работы варианты начальных условий отрабатываются на Земле, преобразуются в файлы обновления, передаваемого на борт. Имеется также возможность их оперативного ввода непосредственно на борту при

работе со специальной панелью модуля интерфейса пользователя. К таким начальным условиям относятся конкретные сценарии работы (цепочки предписанных действий), состав иллюстративного материала в зависимости от используемого оборудования, показания приборов, значения даты и времени и т.д.

3. Виртуальные руководства могут быть использованы для первоначального обучения [5], для получения конкретных справочных данных по тем или иным действиям [6], для восстановления в памяти ранее изученного материала [7], для сопровождения реальной работы, для оценки знаний материала оператором [8]. Для этих целей организованы различные режимы работы программного модуля: произвольный последовательный просмотр операций, автоматическая презентация процессов в рекомендуемом темпе, воспроизведение ранее записанной авторизованной сессии, интерактивный режим работы с виртуальными моделями оборудования.

4. Реализована наземная статистическая обработка полученных результатов в различных аспектах, с графической интерпретацией авторизованных протоколов. При этом анализируются электронные анкеты самих космонавтов, заполняемые после проведения работы, – так называемые листы регистрации, и основной log-файл – пошаговый протокол проведения операций по заданному сценарию.

5. Реализация интерактивного режима и ввод начальных условий обеспечиваются под управлением специального встроенного «сценарного языка описания космических экспериментов», с помощью которого на Земле создаются управляющие XML-файлы частных сценариев.

6. Для обеспечения интерактивного взаимодействия космонавта с информационной средой в иллюстративный материал виртуальных руководств вводятся сенсорные зоны. Оператор воздействует на эти зоны на экране планшета и задает последовательность действий. Виртуальная среда отвечает запрограммированной реакцией, одновременно воспроизводятся мультимедийные информационные ресурсы. Все правильные и ошибочные действия оператора протоколируются для последующего анализа.

В данное время завершается работа над бортовой планшетной версией персонального терминала космонавта, которая подключается для обновления данных к бортовому серверу RSK2 по сети wi-fi.

Работа космонавта полностью авторизована, персональные сценарии передаются ему в соответствии с индивидуальной ролью в коллективных экспериментах [9]. Дальнейшее расширение функциональных возможностей программно-аппаратного комплекса информационной поддержки космических экспериментов существенно повысит удобство, наглядность, оперативность действий



Рис. 1. Главный формат интерфейса ВИРУ в режиме медиа

Fig. 1. The main VIRU interface format in media mode

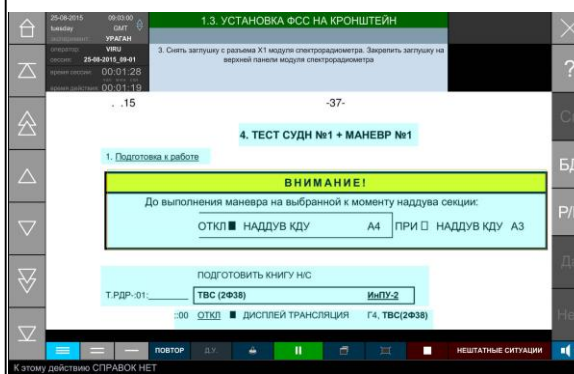


Рис. 2. Экран ВИРУ с фрагментом классической страницы бортовой документации

Fig. 2. VIRU screen with a fragment of an onboard documentation classical page

операторов, обеспечит высокую скорость доступа к материалам бортовой документации.

Специфической особенностью информационной поддержки космических экспериментов является необходимость параллельного сопровождения действий экипажа двумя сюжетными линиями (рис. 2) [10].

С одной стороны, это заранее подготовленная сценарная последовательность, моделирующая виртуальную среду эксперимента, с другой – классический массив бортовой документации.

Ключевые точки виртуального процесса при этом должны быть однозначно взаимосвязаны с пунктами документации. Для реализации такой взаимосвязи предусматриваются органы управления, позволяющие быстро переключать тип доступа к информации, осуществлять переходы по гиперссылкам, синхронизировать динамические процессы и обеспечивать непрерывность обеих линий информационной поддержки.

Литература

1. Операции и эксперименты на Международной космической станции 25 апреля 2013 года. URL: <http://www.federspace.ru/main.php?id=2&nid=20060> (дата обращения: 15.05.2015).

2. Технические эксперименты и исследования на Российском сегменте МКС. Эксперимент «ВИРУ». URL: <http://www.energia.ru/ru/iss/researches/techn/58.html> (дата обращения: 15.05.2015).

3. Жук Е.И., Гуторов А.М. Применение виртуальных руководств при подготовке к проведению научных исследований на пилотируемых орбитальных комплексах. Научные чтения памяти К.Э. Циолковского, Калуга, 2009. URL: <http://readings.gmik.ru/lecture/2009-primenenie-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemih-orbitalnih-kompleksah> (дата обращения: 15.05.2015).

4. Кравченко С.И. Бортовой модуль информационной поддержки экипажа Международной космической станции // Программные продукты и системы. 2013. № 3 (103). С. 68–72.

5. Душенко А.Г., Арестов Д.С. Многофункциональная информационная система поддержки действий экипажа // Программные продукты и системы. 2013. № 3 (103). С. 54–61.

6. Осипов В.П., Сивакова Т.В., Судаков В.А., Трахтенгерц Э.А., Загребев Б.В. Методологические основы поддержки

принятия решений при планировании научно-прикладных исследований и экспериментов на Международной космической станции (МКС) // Электротехнические и информационные комплексы и системы. 2013. № 3 (9). С. 81–88.

7. Михеев А.М., Семочкина И.Ю. 3d-моделирование в интеллектуальной информационной системе поддержки научного эксперимента для систем мониторинга и контроля изделий ракетно-космической техники // Надежность и качество сложных систем. 2013. № 4. С. 28–32.

8. Забавникова Т.Ю. Элементы эргономики проблеме проектирования интерфейса // Вестн. Тамбовского ун-та: Сер. Естественные и технические науки. 2009. № 1 (14). С. 227–228.

9. Мельникова Р.В. Проектирование пользовательского интерфейса // Восточ.-Европ. журн. передовых технологий. 2010. № 8 (48). С. 18–20.

10. Быстров Д.А. Архитектуры презентационной системы образовательного насыщенного мультимедиа контента // Образовательные технологии и общество. 2007. № 4 (10). С. 329–338.

DOI: 10.15827/0236-235X.112.055-057

Received 09.09.15

THE INFORMATION SUPPORT FOR SPACE EXPERIMENTS

Stepanov V.V., Leading Engineer; **Andreev D.A.**, Engineer-Programmer; **Gerken E.A.**, Engineer

(Don Branch of the Space Simulator Center,

Platovsky Av. 101, Novocherkassk, 346400, Russian Federation, artstory@bk.ru;

Obydenov S.S., Leading Engineer

(S.P. Korolev Rocket and Space Public Corporation Energia,

Lenina St. 4a, Korolev, Moscow Reg., 141070, Russian Federation, gcn2@sfo.ru)

Abstract. The article presents the results of testing the virtual manuals for space experiments on board the Russian section of the ISS. Based on this experience the authors set a problem of transferring scientific documentation to crew tablets. The paper presents a description of the new onboard interface module with advanced features such as versioning script with multiple levels of difficulty and switching versions during operation; advanced configuration device of initial conditions for more intuitive and accurate reproduction of planned work; affordable choice of sequence and real-time input of the initial conditions on board. Therefore, there is a variation of operating scenarios and chains of the prescribed actions, the composition of illustrative material and the used equipment, environment virtual models and readings, date and time. There is a specific feature of space experiments information support consisting in the need of crew actions parallel support by two subplots, which are: prearranged scenario sequence simulating a virtual environment for an experiment, and random access to a classic array of board documentation. In this case key points of the virtual process should be clearly linked to the document items. The paper also describes system solutions and a methodology of information space interactive management. The attention is paid to operator's situational behavior, virtual controls work, results logging and crew evaluation. It is emphasized that the astronaut's work is fully authorized on the level of personal scenarios that are transferred in accordance with his individual role in the collective experiments. The authors predict further extension of functionality of hardware and software package of space experiments information support, as well as a significant increase in convenience, visibility, operators' efficiency, obtaining high speed of access to onboard documentation.

Keywords: space experiment, virtual manual, multimedia, Russian segment of ISS, modeling, program cycles, interface.

References

1. *Operatsii i eksperimenty na Mezhdunarodnoy kosmicheskoy stantsii 25 aprelya 2013 goda* [Operations and experiments at the ISS on April 25, 2013]. Available at: <http://www.federspace.ru/main.php?id=2&nid=20060> (accessed May 15, 2015).

2. *Technical experiments and researches on the Russian segment of ISS. "VIRU" experiment*. Available at: <http://www.energia.ru/ru/iss/researches/techn/58.html> (accessed May 15, 2015).

3. Zhuk E.I., Gutorov A.M. Using of the virtual managements for preparation to scientific researches on piloted orbital complexes. *Nauchnye chteniya pamyati K.E. Tsiolkovskogo* [Scientific readings in memory of K.E. Tsiolkovsky]. Kaluga, 2009. Available at: <http://readings.gmik.ru/lecture/2009-primenenie-virtualnih-rukovodstv-pri-podgotovke-k-provedeniyu-nauchnih-issledovaniy-na-pilotiruemih-orbitalnih-kompleksah> (accessed May 15, 2015).

4. Kravchenko S.I. The onboard module for information support of the international space station crew. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 3 (103), pp. 68–72 (in Russ.).

5. Dushenko A.G., Arestov D.S. Multifunctional crew activity support information system. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 3 (103), pp. 54–61 (in Russ.).

6. Osipov V.P., Sivakova T.V., Sudakov V.A., Trakhtengerts E.A., Zagreb B.V. Methodological base of decision-making support when planning scientific and applied research and experiments at the International Space Station (ISS). *Elektrotehnicheskie i informatsionnye komplekсы i sistemy* [Electrical and data processing facilities and systems], 2013, no. 3 (9), pp. 81–88 (in Russ.).

7. Mikheev A.M., Syomochkina I.Yu. 3d-modeling in the science experiment support intelligent information system for rocket-and-space monitoring and control system. *Nadezhnost i kachestvo slozhnykh sistem* [Quality and Reliability of Complex Systems], 2013, no. 4, pp. 28–32 (in Russ.).

8. Zabavnikova T.Yu. Ergonomics elements for the interface design problem. *Vestnik Tambovskogo universiteta. Seriya: Estestvennye i tekhnicheskie nauki* [Tambov Univ. Reports. Series: Natural and Technical Sciences], 2009, no. 1 (14), pp. 227–228 (in Russ.).

9. Melnikova R.V. User interface design. *Vostochno-Evropeysky zhurnal peredovykh technology* [Eastern-European Journal of Enterprise Technologies], no. 8 (48), pp. 18–20 (in Russ.).

10. Bystrov D.A. Architectures of a learning multimedia content presentation system. *Obrazovatelnye tekhnologii i obshchestvo* [Educational Technology and Society], 2007, no. 4 (10), pp. 329–338 (in Russ.).

УДК 004.02

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.058-061

СРАВНИТЕЛЬНЫЙ АНАЛИЗ И КЛАССИФИКАЦИЯ ЗАДАЧ ПО ОПТИМАЛЬНОМУ РАСПОЛОЖЕНИЮ ГРУЗОВ В ОГРАНИЧЕННОМ ПРОСТРАНСТВЕ

А.С. Погорелов, инженер-программист, pogorelov.alserg@gmail.com;

Д.А. Андреев, инженер, dimak134@yandex.ru

*(Донской филиал Центра тренажеростроения,
просп. Платовский, 101, г. Новочеркасск, 346400, Россия);*

А.Н. Панфилов, к.т.н., доцент, panfiloff@rambler.ru

*(Южно-Российский государственный политехнический университет (НПИ) им. М.И. Платова,
ул. Просвещения, 132, г. Новочеркасск, 346428, Россия)*

Работа посвящена проблеме оптимального размещения объектов в ограниченном пространстве. Данная задача относится к классу задач раскроя и упаковки. Дается общая постановка задачи раскроя и упаковки. Описаны подход к классификации задач и критерии, на основе которых строится типология и выделяются основные категории задач. Обзор научных исследований по данной проблеме показывает, что в различных предметных областях возникают похожие задачи, каждая из которых имеет свои особенности, обусловленные спецификой конкретной предметной области. Рассматривается несколько задач по оптимальному расположению грузов, сходных с задачей размещения грузов на борту транспортного грузового корабля. Делается сравнение по основным характеристикам рассмотренных задач с задачей размещения грузов на борту транспортного грузового корабля. Кроме этого, для каждой задачи выявляется класс в рамках рассмотренной типологии, к которому ее можно отнести. В результате сравнительного анализа делается вывод о том, что универсального метода решения задачи оптимального размещения не существует, каждая конкретная практическая задача имеет свои особенности и ограничения, которые необходимо учитывать. Ставится задача разработки нового метода поиска оптимального расположения грузов, учитывающего специфические особенности предметной области, связанной с космическими кораблями.

Ключевые слова: оптимальное расположение груза, транспортный грузовой корабль, задача упаковки, центр масс, космический корабль.

Задача поиска оптимального расположения грузов на борту *транспортного грузового корабля* (ТГК) исследовалась авторами ранее [1]. В рассматриваемом случае в задаче размещения грузов есть определенные особенности:

- форма грузов может быть произвольной;
- множество контейнеров, в которые укладывается груз, конечно;
- форма грузового отсека сложная и состоит из трех основных частей: нижней полусферы, цилиндрической проставки и верхней полусферы;
- имеется ограничение по центру масс корабля (после загрузки грузового отсека отклонение центра масс корабля от нормы не должно превышать установленного значения).

Данная задача относится к классу задач раскроя и упаковки, постановка которой в обобщенном виде представлена в работе [2]. Рассмотрим общую постановку задачи упаковки.

Имеются два множества элементов: контейнеров и упаковываемых предметов.

Контейнеры и предметы – это одно-, двух-, трех- или N-мерные геометрические объекты. Соответственно, речь идет об одно-, двух-, трех- или N-мерной задаче упаковки.

Требуется выбрать некоторые или все из доступных предметов, сгруппировать их в одно или несколько подмножеств и поставить в соответствие каждому из получившихся подмножеств один из имеющихся контейнеров. В результате все

упаковываемые предметы каждого из подмножеств должны быть расположены в пределах контейнеров так, чтобы выполнялись следующие условия: все предметы должны быть расположены внутри контейнеров и упаковываемые предметы не должны пересекаться.

Кроме этого, в задаче присутствует одномерная или многомерная целевая функция, значение которой необходимо оптимизировать.

Общую задачу упаковки можно условно разделить на пять подзадач:

- выбор контейнеров;
- выбор упаковываемых предметов;
- группировка выбранных предметов на подмножества;
- нахождение соответствия между подмножествами предметов и множеством контейнеров;
- размещение предметов в каждом из выбранных контейнеров с соблюдением геометрических ограничений.

В работе [2] также описывается новый подход к классификации задач раскроя и упаковки. На основе пяти критериев строится типология и выделяются основные категории задач. Это дает возможность систематизации научных исследований по проблеме раскроя и упаковки. Кроме этого, разработанная типология имеет большую практическую значимость: классифицируя конкретную задачу, можно отнести ее к какой-либо категории, что позволяет более подробно изучить ее особенности,

связанные с данной категорией, а изучив методы решения задач, относящихся к одной категории, можно выбрать наиболее подходящие аспекты и адаптировать их для решения конкретной поставленной задачи.

Выделяют пять критериев классификации задач раскроя и упаковки [2].

1. Размерность задачи. Задача может быть одно-, двух-, трех- или N-мерной.

2. Оптимизируемое свойство. Выделяются два случая:

- объем контейнеров фиксированный; требуется максимизировать суммарный объем размещаемых предметов, то есть разместить в контейнерах максимум предметов;

- объем размещаемых предметов фиксированный; требуется минимизировать пространство, куда размещаются предметы, то есть разместить все имеющиеся предметы в минимально возможном объеме внутри контейнеров.

3. Вид размещаемых предметов. Выделяют следующие предметы:

- однородные (предметы одной формы и размеров);

- слабооднородные (все предметы можно сгруппировать в относительно небольшое число классов, содержащих однородные предметы);

- неоднородные (среди всего множества предметов есть лишь небольшое число однородных предметов, почти все предметы различны по форме и размерам).

4. Множество контейнеров. Различают задачи с одним контейнером для упаковки и задачи с несколькими контейнерами.

5. Форма размещаемых предметов, которая может быть правильной (прямоугольники, шары, параллелепипеды, цилиндры и т.д.) и неправильной.

В зависимости от комбинации описанных критериев для конкретной задачи она будет отнесена к тому или иному классу задач раскроя и упаковки.

Описываемую в работе [1] задачу поиска оптимального расположения грузов на борту транспортного грузового корабля можно отнести к промежуточному классу MHLOPP (Multiple heterogeneous large object placement problem – задача размещения во множестве неоднородных контейнеров).

Обзор научных исследований по проблеме оптимального размещения объектов в ограниченном пространстве показывает, что в различных предметных областях возникают похожие задачи, имеющие каждая свои особенности, связанные со спецификой конкретной предметной области.

Рассмотрим несколько задач по оптимальному расположению грузов, которые имеют сходство с задачей размещения грузов на борту транспортного грузового корабля.

В работах [3, 4] рассматривается действующая программная система *packer3d*, предназначенная

для расчета оптимальной укладки грузов в транспортные средства. В этой системе грузы, как и объем, в который они помещаются, имеют форму параллелепипеда с заданными размерами. Требуется рассчитать точное положение ящиков в заданном объеме таким образом, чтобы его заполнение грузом было допустимым и наиболее эффективным по объему, то есть суммарный объем помещившихся ящиков должен быть максимальным из всех возможных. Кроме этого, алгоритм, описываемый в работе, позволяет учитывать такие дополнительные ограничения, как грузоподъемность, максимальное давление на верхнюю грань ящика, сбалансированность давления на оси транспортного средства [3].

По классификации из работы [2] данную задачу можно отнести к промежуточному классу SLOPP (Single large object placement problem – задача размещения в одном контейнере).

Алгоритм *packer3d* основан на сведении трехмерной задачи упаковки к двумерной и включает в себя следующие части:

- алгоритм комбинирования предметов;

- алгоритм трехмерной задачи (сведение к двумерной);

- выбор области укладки для трехмерного алгоритма;

- алгоритм двумерной укладки.

В этом алгоритме для нахождения наилучшего решения используется эвристический подход, так как точные методы требуют неприемлемо долгого времени расчета. В основе этого подхода лежит определение некоторого функционала качества, который используется для определения оптимального решения. Для подбора оптимального набора характеристик функционала качества используются такие математические методы, как генетические алгоритмы и нейронные сети.

В работах [5, 6] рассматривается математическая четырехмерная модель размещения груза в трюмах судна. Эта модель, как и система *packer3d*, оперирует с грузами и контейнерами (в данном случае трюмами), описываемыми в форме параллелепипеда. Кроме этого, модель способна работать с более чем одним контейнером (трюмом), а также учитывать такие ограничения, как совокупность зон трюмов, запретных для размещения грузов (строительные конструкции, служебные помещения, стационарное оборудование, дороги и т.д.), зазор между соседними грузовыми объектами [6].

По классификации из работы [2] данную задачу можно отнести к промежуточному классу MILOPP (Multiple identical large object placement problem – задача размещения во множестве однородных контейнеров).

Для решения задачи размещения груза в трюмах судна используется генетический алгоритм, причем задача сводится к двумерной путем разбиения грузов на стопки одинаковой высоты.

Также в алгоритме используется критерий эффективности, накладывающий дополнительные ограничения на параметры задачи.

В работе [7] предлагаются математическая модель и метод решения 3D-задачи оптимальной упаковки цилиндров и параллелепипедов в параболический контейнер с круговыми стеллажами с учетом минимально допустимых расстояний и ограничений поведения механической системы (динамическое равновесие, моменты инерции, устойчивость). При этом задача состоит в том, чтобы разместить множество элементов на полках внутри отсеков контейнера с учетом минимально допустимых расстояний и ограничений поведения системы так, чтобы расстояние между центром тяжести множества загруженных элементов и заданным центром тяжести контейнера достигало своего минимального значения.

Особенность этой задачи в отличие от предыдущих в том, что в ней грузы могут иметь форму не только параллелепипеда, но и цилиндра, а контейнер для загрузки имеет форму параболоида. Кроме этого, в число ограничений входит ограничение по центру тяжести.

По классификации из работы [2] данную задачу можно отнести к промежуточному классу MHLOPP (Multiple heterogeneous large object placement problem – задача размещения во множестве неоднородных контейнеров).

Описываемая математическая модель использует ϕ -функции Стояна [8], которые позволяют описывать математические модели задач упаковки в виде задач нелинейной оптимизации с целью применения для их решения методов локальной и глобальной оптимизации.

В работе [9] рассматривается задача упаковки грузов в заданном объеме на примере их упаковки в самолет с известным центром тяжести. Для обеспечения нормального маневра самолета грузы необходимо уложить в имеющийся в самолете контейнер так, чтобы отклонение от известного центра тяжести самолета было минимальным.

По классификации из работы [2] данную задачу можно отнести к промежуточному классу SLOPP (Single large object placement problem – задача размещения в одном контейнере).

Задача формулируется в виде оптимизационной задачи обратно-выпуклого программирования,

для решения которой используется численный метод. Приводятся результаты решения тестовых примеров.

В работе [10] рассматривается подход к оптимизации положения центра масс набора блоков. Этот подход является общим и не привязан к какой-либо конкретной предметной области. Особенность его в том, что оптимизация положения центра масс осуществляется для набора уже упакованных грузов, то есть исходным для алгоритма является результат оптимального размещения грузов по какому-либо другому методу упаковки, не учитывающему центр масс. Однако существенным недостатком данного подхода является необходимость низкой плотности первоначальной упаковки грузов, когда объем контейнера превышает суммарный объем всех блоков и остается достаточно места для перемещения.

По классификации из работы [2] данную задачу можно отнести к промежуточному классу SLOPP (Single large object placement problem – задача размещения в одном контейнере).

Алгоритм является итеративным. Положение центра масс набора блоков вычисляется повторно после сдвига каждого блока. Сдвигаемые на каждом шаге блоки выбираются не случайно, а в определенном порядке, позволяющем оптимизировать положение центра масс за один проход по каждой из координат.

В таблице представлено сравнение по основным характеристикам рассмотренных задач с задачей размещения грузов на борту транспортного грузового корабля. Символ «+» означает совпадение характеристик, символ «-» – отличие задач друг от друга в отношении данной характеристики, символ «±» – частичное совпадение характеристик.

Рассмотрев различные варианты практических задач по оптимальному расположению грузов и выделив сходства и различия между ними и задачей оптимального размещения грузов на борту транспортного грузового корабля, можно сделать вывод, что универсального метода решения задачи оптимального размещения не существует, в каждой конкретной задаче есть свои особенности и ограничения, которые необходимо учитывать. С другой стороны, между рассмотренными задачами существуют сходства, что позволяет использовать определенные приемы решения для похо-

Результаты сравнения задач

Task comparison results

Задача/ алгоритм/ метод	Характеристики			
	Форма грузов	Форма грузового отсека	Количество контейнеров	Ограничение по центру масс
Алгоритм packer3d	±	–	–	–
Математическая модель размещения груза в трюмах судна	±	–	+	–
Метод оптимальной упаковки в параболический контейнер	±	±	±	+
Задача упаковки грузов в самолет с известным центром тяжести	±	–	–	+
Метод оптимизации положения центра масс набора блоков	±	–	–	±

жей задачи. Учитывая специфику предметной области, связанной с космическими кораблями, для задачи оптимального размещения грузов на борту транспортного грузового корабля необходимо разработать новый метод решения, основанный на уже проверенных методах и учитывающий специфические особенности рассматриваемой задачи.

Литература

1. Погорелов А.С., Панфилов А.Н., Андреев Д.А. Задача оптимального размещения грузов на борту транспортного грузового корабля // Инженерный вестн. Дона. 2015. № 3; URL: <http://www.ivdon.ru/ru/magazine/archive/n3y2015/3140> (дата обращения: 08.09.2015).
2. Wäscher G., Haußner H., Schumann H. An improved typology of cutting and packing problems. *European Journ. of Operational Research*, 2007, no. 183, pp. 1109–1130.
3. Псиола В.В. О приближенном решении 3-мерной задачи об упаковке на основе эвристик // Интеллектуальные системы. Теория и приложения. 2007. № 1–4. С. 83–100.
4. Псиола В.В. Оценка качества алгоритма Packer3D (Теория и практика). М.: Изд-во МГУ, 2009. 35 с.

5. Нырклов А.П., Соколов С.С. Эффективное математическое и алгоритмическое обеспечение оперативного размещения груза на транспорте // Наука в жизни современного человека: матер. Междунар. симпози. 2013. URL: www.sworld.com.ua (дата обращения: 08.09.2015).
6. Соколов С.С. Математическая модель рационального размещения груза в трюмах судна // Вестн. гос. ун-та морского и речного флота им. адмирала С.О. Макарова. 2010. № 3. С. 89–92.
7. Коваленко А.А., Панкратов А.В., Романова Т.Е. Размещение объектов в контейнере параболической формы с круговыми стеллажами с учетом ограничений поведения // Журн. выч. и приклад. матем. 2013. № 2. С. 23–32.
8. Chernov N. Phi-Functions for 2D Objects Formed by Line Segments and Circular Arcs. *Advances in Operations Research*. 2012. URL: <http://dx.doi.org/10.1155/2012/346358> (дата обращения: 08.09.2015).
9. Остапенко В.В., Соболенко Л.А., Прохорович И.В. Метод обратно-выпуклого программирования и оптимальная упаковка грузов // Системные исслед. и информ. техн. 2004. № 2. С. 95–103.
10. Карпов К.Э., Яновский В.В. Offline-алгоритм оптимизации центра масс упакованных блоков // Изв. СПб ГЭИ ЛЭТИ. 2007. № 9. С. 10–16.

DOI: 10.15827/0236-235X.112.058-061

Received 09.09.15

COMPARISON ANALYSIS AND TASK CLASSIFICATION ACCORDING TO CARGO OPTIMAL PLACEMENT IN RESTRICTED SPACE

Pogorelov A.S., Engineer-Programmer, pogorelov.alserg@gmail.com; **Andreev D.A.**, Engineer, dimak134@yandex.ru (Don Branch of the Space Simulator Center, Platovsky Av. 101, Novocherkassk, 346400, Russian Federation);

Panfilov A.N., Ph.D. (Engineering), Associate Professor, panfiloff@rambler.ru (South-Russian State Polytechnical University, Prosveshcheniya St. 132, Novocherkassk, 346428, Russian Federation)

Abstract. The paper is devoted to the problem of objects optimal placement in restricted space. This problem belongs to the class of cutting and packing problems. The paper formulates the common cutting and packing problem. It also describes the approach to such problems classification and criteria, which are the basis of the typology and the main categories of the problems. The scientific researches review shows that similar problems arise in various domains and these problems have various features as a specificity of a problem domain. The authors consider several problems of cargo optimal placement, which have some similarities with the problem of cargo placement on a transport spacecraft. They compare considered problems to the problem of cargo placement on the transport spacecraft according to common characteristics of these problems. Furthermore, each problem under consideration is assigned to one of the classes in terms of the given typology. As a result of a comparison analysis, there is a conclusion that there is no any universal decision method for the optimal placement problem. There are some particular features and restrictions for any real problem, which must be considered. The authors set a task of development of a new method to find the optimal cargo placement. This method must consider any specific features of the problem domain related to spacecrafts.

Keywords: cargo optimal placement, cargo spacecraft, cutting and packing problem, center of mass, spacecraft.

References

1. Pogorelov A.S., Panfilov A.N., Andreev D.A. The problem of finding the cargo optimal placement on a cargo spacecraft board. *Inzhenerny vestnik Dona* [Engineering Journ. of Don]. 2015, no. 3 (in Russ.). Available at: <http://www.ivdon.ru/ru/magazine/archive/n3y2015/3140> (accessed September 08, 2015).
2. Wäscher G., Haußner H., Schumann H. An improved typology of cutting and packing problems. *European Journ. of Operational Research*. 2007, no. 183, pp. 1109–1130.
3. Psiola V.V. On an approximate solution of 3-dimensional packing problem based on heuristics. *Intellektualnye sistemy. Teoriya i prilozheniya* [Intelligent system. Theory and applications]. 2007, no. 1–4, pp. 83–100 (in Russ.).
4. Psiola V.V. *Otsenka kachestva algoritma Packer3D (Teoriya i praktika)* [Packer3D Algorithm Quality Assessment (Theory and Practice)]. Moscow, Moscow State Univ. Publ., 2009, 35 p.
5. Nyrkov A.P., Sokolov S.S. *Effektivnoe matematicheskoe i algoritmicheskoe obespechenie operativnogo razmesheniya gruzha na transporte* [Effective mathematical and algorithmic support of the cargo operational placement in transport]. State Univ. of sea and river fleet n.a. Admiral S.O. Makarov. Available at: <http://www.sworld.com.ua/index.php/ru/c113-4/16239-c113-012> (accessed September 08, 2015).
6. Sokolov S.S. A mathematical model of rational placement of cargo in the holds of the ship. *Vestnik gos. univ. morskogo i rechnogo flota im. admirala S.O. Makarova* [Bulletin of the State Univ. of Sea and River Fleet n.a. Admiral S.O. Makarov]. 2010, no. 3, pp. 89–92 (in Russ.).
7. Kovalenko A.A., Pankratov A.V., Romanova T.E. Object placement in a parabolic shape container with circular racks with regard to behavior limits. *Zhurnal vychislitel'noy i prikladnoy matematiki* [Journ. of Computational and Applied Mathematics]. 2013, no. 2, pp. 23–32 (in Russ.).
8. Chernov N. Phi-Functions for 2D Objects Formed by Line Segments and Circular Arcs. *Advances in Operations Research*. 2012; Available at: <http://www.ivdon.ru/ru/magazine/archive/n3y2015/3140> (accessed September 08, 2015).
9. Ostapenko V.V., Sobolenko L.A., Prokhorovich I.V. The inverse-convex programming method and optimal cargo packing. *Sistemnye issledovaniya i informatsionnye tekhnologii* [System Research and Information Technologies]. 2004, no. 2, pp. 95–103 (in Russ.).
10. Karpov K.E., Yanovsky V.V. Offline optimization algorithm of the center of mass of packed blocks. *Izvestiya Sankt-Petersburgskogo gos. elektrotehnicheskogo universiteta "LETI"* [The Bulletin of the St.-Petersburg State Electrotechnical University "LETI"]. 2007, no. 9, pp. 10–16 (in Russ.).

УДК 004(384+31)

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.062-069

ПРОБЛЕМЫ СОЗДАНИЯ ВЫСОКОТЕМПЕРАТУРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

(Работа выполнена при поддержке гранта РФФИ № 14-29-09207 офи_м
«Фундаментальные аспекты конструктивно-схемотехнической организации элементов радиационно-стойких
запоминающих устройств для высокотемпературной микроэлектроники»)

С.Г. Бобков, д.т.н., профессор, зам. директора, bobkov@cs.niisi.ras.ru

(Федеральный научный центр

Научно-исследовательский институт системных исследований РАН, ФНЦ НИИСИ РАН,

Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия;

Национальный исследовательский ядерный университет «МИФИ»,

Каширское шоссе, 31, г. Москва, 115409, Россия)

В статье рассматриваются проблемы создания высокотемпературных электронных компонент для нефтяной, газовой, авиационной и космической отраслей. Рассмотрены пути развития систем Умных скважин. Создание отечественных систем Умных скважин, помимо решения задачи импортозамещения, позволит существенно снизить стоимость оборудования. Так, стоимость электронно-управляемых задвижек вблизи перфорационных отверстий Умной скважины могла бы быть снижена с нескольких сотен тысяч долларов до десятков. Аналогичные проблемы существуют и в космической отрасли: стоимость западного вычислителя космического применения может достигать полумиллиона евро, аналогичный отечественный вычислитель до 10 раз дешевле. Полеты в дальний космос потребуют, помимо высокой радиационной стойкости, функционирования в широком диапазоне температур. В статье представлены технические данные базовых микросхем разработки НИИСИ РАН с температурой функционирования до +125 °C и пути повышения температуры их функционирования. Показана принципиальная возможность создания высокотемпературных микросхем на их основе.

Ключевые слова: высокотемпературная электроника, интеллектуальная скважина, КНИ-технология, система на кристалле, коммутатор SpaceWire, коммутатор Serial RapidIO, микропроцессор.

Под высокотемпературной электроникой понимается электроника, функционирующая при температуре свыше 150 °C. Основные области ее применения: добыча энергоресурсов, промышленность, электроника для автомобилей, электроника для авиации и космоса [1].

В последние годы наблюдается значительный рост производства высокотемпературных компонент, что связано с увеличением объема выпуска автомобилей, самолетов, ракетно-космической техники, с увеличением глубины нефтегазовых скважин и пр. Целый ряд ведущих мировых компаний начали разработку и выпуск высокотемпературных электронных компонент.

В качестве примера можно привести следующие.

1. Микропроцессоры компании Texas Instruments [2]:

– SM320F28335-HT – 32-разрядный сигнальный микроконтроллер семейства C2000, функционирующий в диапазоне температур от –55 до 210 °C, 150 МГц, включающий 12-разрядный 16-канальный АЦП с временем преобразования 80 нс;

– SM320F2812-HT – 32-разрядный микроконтроллер для управления двигателем, 150 МГц, функционирующий в диапазоне температур от –55 до 220 °C.

2. Высокотемпературные драйверы компании Atmel [3], способные функционировать при температурах переходов до 200 °C и температурах окружающей среды до 150 °C. Они поддерживают

от 3 до 6 независимых задающих каскадов и ток до 1,0 А. Микросхемы разработаны с применением SOI-технологии SMART-I.S.™ с размерами порядка 0,8 мкм. Данная технология обеспечивает множество преимуществ, таких как возможность работы при высоких температурах и высоких напряжениях, компактность, улучшенная защита от эффекта защелкивания, низкие токи утечки и другие. Драйверы в сочетании с микроконтроллером и дискретными мощными полевыми МОП-транзисторами позволяют создавать блоки управления двигателем постоянного тока без щеток (BLDC).

3. Продукция компании CISOID [4], основанной в 2000 году и специализирующейся на разработке высокотемпературных электронных компонентов: АЦП, операционные усилители, регуляторы напряжения, транзисторы, генераторы импульсов, драйверы ключей, микросхемы источников питания и пр. Компоненты CISOID сохраняют работоспособность при жестких внешних воздействиях (вибрации, механические удары, перепады температур и пр.) в условиях повышенных температур от –55 до +225 °C. Часть изделий успешно выдержала тестирование NASA при экстремальных температурах от –195 до +400 °C для применения в составе космической аппаратуры. Продукция компании применяется в нефтегазовой, космической и авиационной отраслях, а относительно невысокая стоимость допускает использование в высоконадежных промышленных и автомобильных приложениях.

Высокотемпературная электроника для нефтегазовой отрасли

Для России наиболее критичной областью является нефте- и газодобыча. Актуальность работ по созданию требуемой электроники значительно выросла после введения санкций на поставку таких компонент в Россию.

Наиболее значимой задачей является создание *Умных скважин* (Intelligent Well – IW) [5, 6]. Широко распространенное определение предполагает, что Умной скважиной является та, которая имеет следующие возможности:

- разделение потока: индивидуальные зоны/ответвления в скважине изолированы друг от друга и поток из них или в них может управляться дистанционно посредством устройства управления скважинным потоком;
- контроль параметров скважины: ствол скважины и/или параметры пласта могут контролироваться дистанционно в реальном масштабе времени;
- оптимизация производительности скважины: устройство управления скважинным потоком может регулироваться по результатам знаний, полученных от оценки ствола скважины с целью достижения максимальной производительности данной скважины.

IW обеспечивает дополнительную выгоду, а именно: ускоренное восстановление, снижение себестоимости сооружения скважины, ускорение профилирования добычи, снижение частоты вмешательств в работу скважины и затрат, улучшенную эксплуатационную безопасность.

Условия эксплуатации в скважине экстремальные. Температура скважины более 100 °C, высокое давление, стальные вентили, предназначенные для работы в тяжелых условиях, и сложная обсадка – внутреннее пространство нефтяной или газовой скважины.

Дальнейшее развитие системы IW подтверждает, что эта технология находит все большее признание: так, почти стандартным стало теперь требование о необходимости совместимости морских подводных систем управления и древовидных систем с технологией IW. Однако системы IW являются относительно новым делом, совсем небольшой практический опыт их использования в реальных условиях не позволяет делать традиционные прогнозы надежности.

Самой важной причиной для введения системы IW с целью управления всем производством является необходимость улучшения реальных показателей добычи на разрабатываемых месторождениях. Но подобное возможно только в случае, когда система достигает требуемой надежности и удобства эксплуатации. При определении желательной надежности системы IW следует рассмот-

реть много различных параметров, таких как капитальные затраты на оборудование, использующаяся технология и последствия отказа системы. Эти параметры могут значительно отличаться на разных месторождениях и при разных условиях эксплуатации.

Одной из базисных проблем при оценке надежности систем IW является ограниченный практический опыт в реальных условиях, который осложняет получение важных исходных данных. Следовательно, имеется определенный риск получения неправильных заключений, если внимание фокусируется исключительно на абсолютной надежности прогнозов.

Для улучшения надежности системы управления и мониторинга IW создаваемые компоненты должны быть основаны на архитектуре, в которой заложены избыточность и отказоустойчивые схемотехнические решения. Эти решения должны применяться для создания систем всех типов скважин, включая производящие и нагнетательные.

Важным компонентом системы является система телеметрии скважин, состоящая из наземного и подземного блоков. Ее функциями являются контроль над работой погружного двигателя и мониторинг параметров среды в забое скважины; диагностика неполадок и проведение предупредительных профилактических мероприятий (например засорение, запарафинивание); оптимизация режима работы установки электроцентробежных насосов (работа на максимальной депрессии, оптимальный режим автоматического повторного включения); проведение гидродинамических исследований в постоянно работающих скважинах. Необходимо также обеспечить передачу геофизической информации на поверхность для таких параметров, как температура, давление масла и окружающей среды, вибрация корпуса погружного электродвигателя и др.

Из приведенных требований следует, что для электронной системы IW необходимо создать микропроцессоры со встроенными датчиками, памятью, сетевыми и системными контроллерами. Для высокоинтеллектуальных систем необходимо также создать отдельные микросхемы памяти.

Типовой температурный градиент в скважинах составляет 25 °C на километр глубины, поэтому на больших глубинах температура в скважине может достигать 175 °C. В настоящее время ведется экспериментальная разработка полезных ископаемых на сверхглубинах, где температура может превышать 200 °C, при этом давление в таких скважинах составляет 25 кПа/дюйм. На рисунке 1 приведен пример электронного корпуса для проведения телеметрии в буре для таких скважин.

Компания Schlumberger разделяет скважины высокого давления и высоких температур на три класса: 150–205 °C, 205–260 °C и свыше 260 °C.

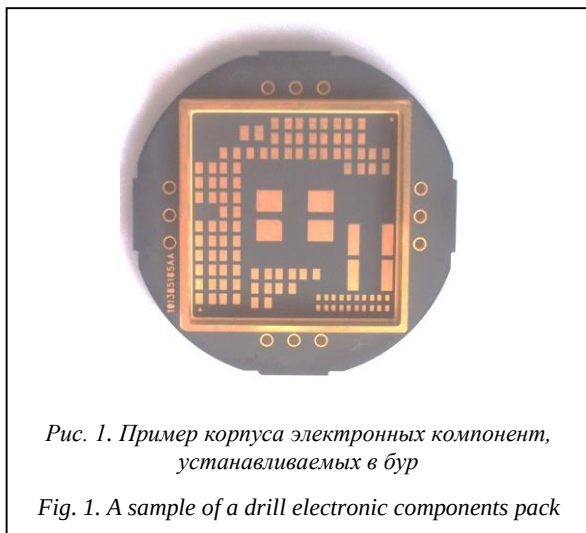


Рис. 1. Пример корпуса электронных компонент, устанавливаемых в бур

Fig. 1. A sample of a drill electronic components pack

Помимо решения важных производственных задач, высокотемпературные вычислительные системы должны обеспечить импортозамещение и существенную экономию средств. Например, каждая из электронно-управляемых задвижек вблизи перфорационных отверстий IW оснащена датчиками положения задвижки, притока флюидов, температур. В 2001 году типичная цена подобной задвижки приближалась к \$500 000.

Высокотемпературная электроника для авионики и космонавтики

Кроме газо- и нефтедобывающей промышленности, высокотемпературные интегральные микросхемы (ИМС) широко используются в авионике и космонавтике [7]. В соответствии с программой *Электронный самолет* (More Electric Aircraft – MEA) в новых проектах вместо центрального компьютера используется множество распределенных микроконтроллеров и микропроцессоров. Это позволяет значительно уменьшить число и длину медных кабелей, снизить вес самолета и увеличить надежность управления, приблизив микроконтроллер к исполнительному механизму. Кроме того, замена центрального компьютера множеством распределенных микроконтроллеров позволяет уменьшить число разъемных соединений, повысить надежность и привести ее в соответствие с требованиями стандарта MIL-HDBK-217F [8]. Однако такое решение (приближение ИМС к двигателям) приводит к тому, что диапазон температур, в котором должны работать ИМС, составляет от –55 до 200 °С. Применение принудительного охлаждения ИМС в таких системах нежелательно, так как это приводит к увеличению веса и стоимости самолета. Кроме того, выход из строя системы охлаждения может вызвать отказ системы управления. Другим аспектом программы MEA является замена гидравлических систем управления электронными,

благодаря чему повышается надежность и снижается стоимость этих систем. Новая технология управления получила название «мехатроника». При ее внедрении необходимо использовать сенсоры, сенсорные интерфейсы, системы обработки данных, которые будут расположены вблизи исполнительных механизмов и, как правило, работают в диапазоне высоких температур, а сами являются источниками тепла. Еще одной областью применения высокотемпературных ИМС является автомобильная электроника. Аналогично авиационной промышленности автомобильная отрасль тоже переходит к замене гидравлических и механических систем управления электронными. Использование электронных систем управления в автомобиле упрощает производство, управление, тестирование автомобиля, что, в конечном счете, позволяет снизить его стоимость и повысить надежность. Особо следует отметить электромобили и гибридные автомобили, в которых управление двигателями обеспечивается электронными системами управления.

Космическое направление высокотемпературной электроники является наиболее наукоемким в силу повышенных требований к надежности и радиационной стойкости. Каждый спутник для научных исследований уникален и во многом из-за этого приходится под каждый эксперимент создавать уникальную вычислительную систему. Специфика области не позволяет использовать коммерческие микросхемы для создания вычислительных систем, серийность требуемых микросхем небольшая, а сами микросхемы должны функционировать в течение многих лет под воздействием радиации. Кроме того, космическое пространство в ряде случаев создает значительные проблемы по отводу тепла. Планируемые полеты в дальний космос неизбежно поставят задачи дальнейшего повышения надежности вычислительных систем и расширения температурного диапазона функционирования.

Основные требования к элементной базе космического применения

Применение микроэлектронных изделий в условиях космического пространства предъявляет к ним ряд специальных требований: долговечность – не менее 200 тысяч часов; стойкость к накопленной дозе – не ниже 200 крад; стойкость к *тяжелым заряженным частицам* (ТЗЧ) с *линейной передачей энергии* (ЛПЭ) – не ниже 100 МэВ×см²/мг; сохранение работоспособности после 50 нс импульсного облучения – до 10¹² рад/с; стойкость к нейтринному потоку – до 10¹³ нейтрино/см²; минимальная чувствительность к сбоям, вызываемым космическими частицами с высокими энергиями, то есть речь должна вестись о создании специальной категории

микросхем для космических применений. Целесообразность разработки такой категории изделий подтверждена практическим опытом ряда фирм, таких как Aeroflex, Atmel и др. Так, для построения бортовых вычислителей наиболее широко используются специальные микропроцессоры, выпускаемые компанией Aeroflex (66 МГц, технология 0,25 мкм, 5 млн транзисторов) и фирмой Atmel (100 МГц, технология 0,18 мкм, 8 млн транзисторов). Оба процессора имеют уровень стойкости к накопленной дозе 300 крад, порог по сбоеустойчивости 10–15 МэВ и уровень стойкости к эффекту защелки 100 МэВ×см²/мг. Для повышения устойчивости микропроцессоров к эффектам одиночных сбоев использованы специальные схемные решения функциональных элементов, а для обеспечения защиты вычислительного процесса от ошибок в процессорах применено мажорирование. В ближайшей перспективе (3–5 лет) указанные фирмы планируют начать выпуск микропроцессоров с рабочей частотой 150–200 МГц и количеством транзисторов, превышающим 10 млн. Таким образом, очевидно, что обеспечение паритета России в использовании космического пространства возможно только при использовании элементной базы такого же уровня.

Дополнительными могут быть требования к широкому температурному диапазону, порой заметно отличающиеся в зависимости от места расположения аппаратуры. В настоящее время в мире существуют 4 общепринятых диапазона темпера-

тур функционирования микросхем: коммерческий – от 0 до +70 °С, промышленный – от –40 до +85 °С, авиационный – от –40 до +125 °С, военный (в рамках стандарта MIL-STD-883) и космический (Space) – от –55 (–60° для России) до +125 °С. Выделяются также высокотемпературные микросхемы, функционирующие при температурах свыше 150 °С.

Часть сложных физических детекторов для исследования космоса не могут функционировать в широком диапазоне температур, соответственно, требования к аппаратуре, устанавливаемой вблизи таких физических установок, аналогичны. Однако в большинстве случаев требования к комплектующим соответствуют понятиям Military и Space. Прежде всего это требования к электронике аппаратов, функционирующих на высоких орбитах и в дальнем космосе.

Одна из наиболее сложных задач – создание электронной аппаратуры для контроля и управления двигателем для полета в дальний космос. Использование множества распределенных микроконтроллеров и микропроцессоров позволит значительно уменьшить число и длину медных кабелей, снизить вес аппарата и увеличить надежность управления. Однако температура функционирования микросхем в подобных условиях приближается к +200 °С.

Таким образом, для таких применений должна создаваться радиационно стойкая высокотемпературная элементная база.

Сравнение микропроцессоров КОМДИВ рад. стойкого применения

A comparison of KOMDIV radiation resistance microchips

Параметр	Микропроцессор						
	5890BE1	5890BM1T	1900BM1T	1907BM014	1907BM038	1907BM048	1907BM028
Частота, МГц	33	33	66	100	125	66	150
Архитектура	КОМДИВ32						КОМДИВ64
Конвейер, стадии	5	5	5	8	8	8	7
Встроенный системный контроллер	да	нет	нет	да	да	да	да
Контроллер интерфейса ГОСТ Р 52070-2003	нет	нет	нет	да	нет	да	нет
Троирование/дублирование	нет	нет	да	нет	нет	да	нет
Температура среды, °С	от –60 до +85		от –60 до +125				
Нормы изготовления, мкм	0,5 КНИ	0,5 КНИ	0,35 КНИ	0,25 КНИ	0,25 КНИ	0,25 КНИ	0,25 КНИ
Потребляемая мощность, Вт	2,4	1,7	3	5	6	7	5,5
Размер L2 кэш-памяти	–	–	–	–	–	–	256 Кбайт
Сопроцессор DSP	нет	нет	нет	нет	да	нет	нет
Число Serial RapidIO 4X	–	–	–	–	1	–	1
Число Ethernet	–	–	–	1	–	–	1
Число Manchester	–	–	–	2	–	2	–
Число SpaceWire	–	–	–	3	4	3	–
Начало серийного выпуска	2009 г.	2011 г.	2012 г.	2016 г.	2016 г.	2016 г.	2016 г.

Основные характеристики базовых микросхем разработки НИИСИ РАН

В таблице дается сравнение основных характеристик микропроцессоров КОМДИВ [9] рад. стойкого применения.

В микросхемах для создания многопроцессорных систем используются стандарты SpaceWire и RapidIO. Они имеют существенно разные скорости передачи и дополняют друг друга. Каналы SpaceWire являются стандартом де-факто для космических применений, однако имеют скорость передач до 200 Мб/с. В стандарте RapidIO предусмотрена скорость передач до 6,25 Гбит/с на линию.

Рассмотрим основные микросхемы, позволяющие создавать законченные вычислительные системы [10].

Система на кристалле (СнК) 1907BM038 является перспективной микросхемой для создания различных систем бортового космического применения. Технология изготовления – КНИ 0,25 мкм. Структурная схема СнК приведена на рисунке 2.

В состав системы входят следующие основные блоки: 32-разрядное управляющее ядро (CPU), 128-разрядный арифметический сопроцессор (на рисунке обозначен как CP2), системный контроллер с набором интерфейсов: DDR2, SPI, два канала

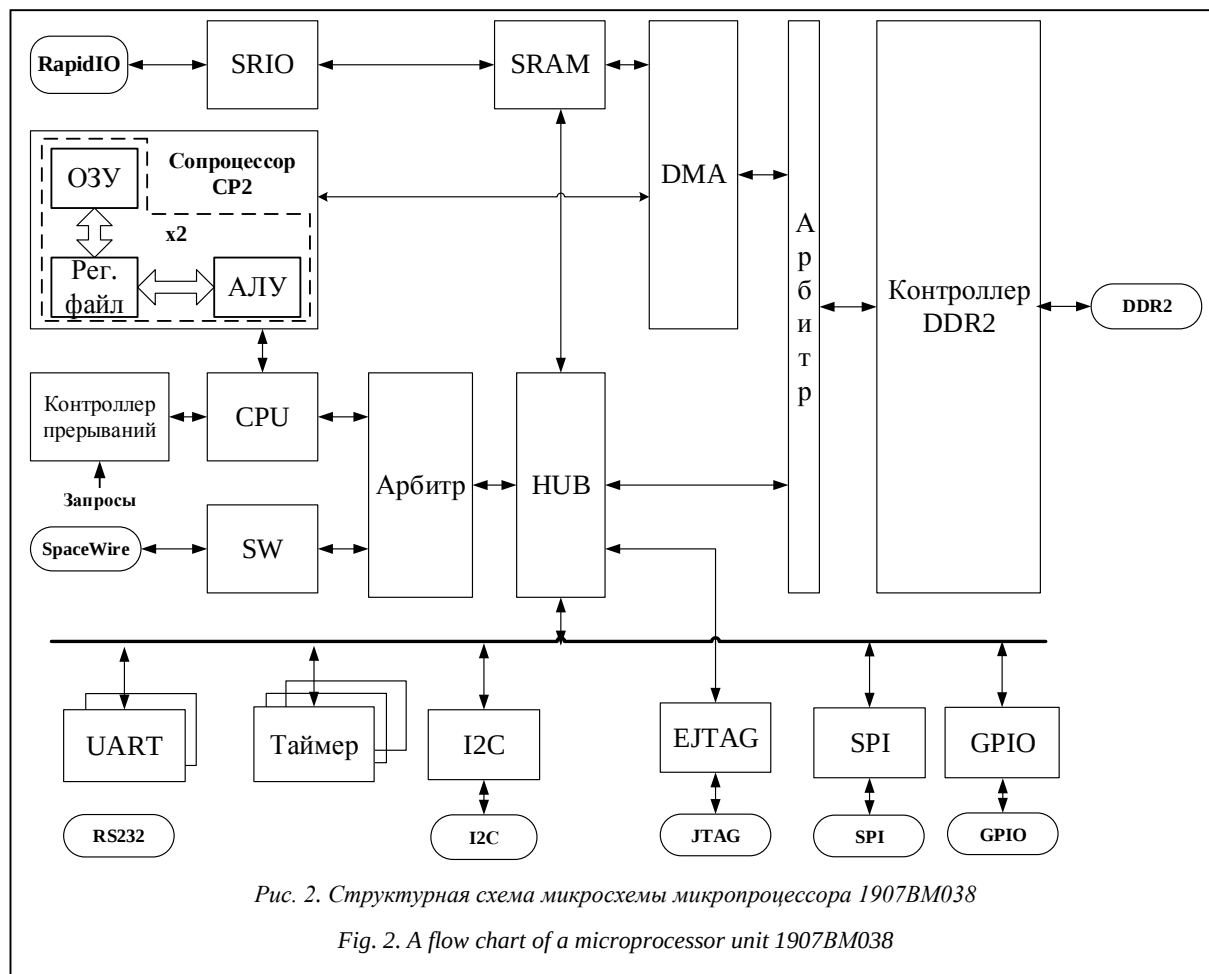
Serial RapidIO, четыре канала SpaceWire, I2C, GPIO, два контроллера UART.

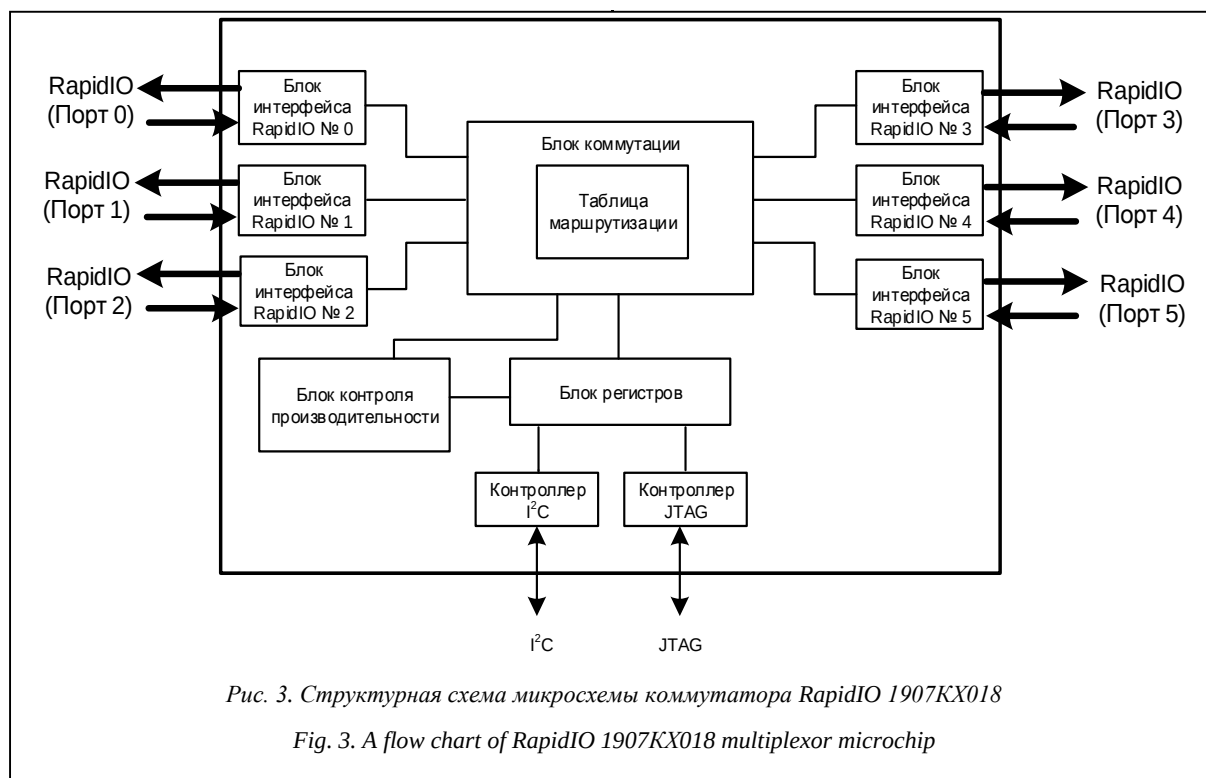
Тактовая частота микросхемы – 100 МГц. Пропускная способность ОЗУ – 512 Мбайт/с.

Конструкция корпуса микросхемы: металлокерамический корпус типа dumped BGA с числом выводов 675 шт.

Микросхема коммутатора последовательных каналов 1907KX018 предназначена для организации связи между устройствами (коммутаторами и СнК), имеющими в своем составе каналы последовательного RapidIO. Коммутатор является перспективной микросхемой для создания высокопроизводительных систем бортового космического применения. Технология изготовления – КНИ 0,25 мкм.

Коммутатор содержит шесть портов последовательного RapidIO, среда передачи – LP-Serial 4X либо 1X. Максимальная скорость передачи по каждой линии – 1,25 Гбит/с. Структурная схема микросхемы представлена на рисунке 3. Отличительными характеристиками являются отдельная таблица коммутации для каждого порта, система контроля производительности, встроенный блок управления ошибками. Коммутатор способен непосредственно соединять до 256 оконечных устройств (СнК, контроллеры ввода-вывода и т.п.) в системе, что обычно является достаточным для





бортовых систем. Индивидуальные таблицы коммутации позволяют гибко настраивать пути прохождения пакетов по сети. Система контроля производительности используется для определения характеристик потока данных в канале, определения перегрузок в сети, определения мест блокировок. Определение величины потока данных в канале производится при выборе топологии сети на этапе моделирования и проведении натурного эксперимента. Причиной перегрузок в сети может быть ухудшение состояния канала, возникшее в результате сбоев. Поэтому раннее оповещение об ошибках уменьшает время, необходимое на подключение резервного пути. Для выбора обходного пути крайне полезной является способность коммутатора определять заблокированные каналы.

Максимальная потребляемая мощность при использовании всех шести портов в режиме 4X не более 6 Вт. Предусмотрено снижение потребляемой мощности в режимах работы портов 1X, а также путем индивидуального отключения передатчиков каждого порта. Микросхема имеет два канала питания: 2,5 В и 3,3 В.

Конструкция корпуса микросхемы: керамический корпус типа dumped BGA с числом выводов 399 шт.

Повышенная стойкость к сбоям достигается комплексом дополнительных мер: помехоустойчивое кодирование при обращении к памяти буферов пакетов; использование резервирования таблиц маршрутизации; топологические и схемотехнические решения библиотечных элементов.

Коммутатор имеет расширенные возможности диагностики работоспособности. Состояние диа-

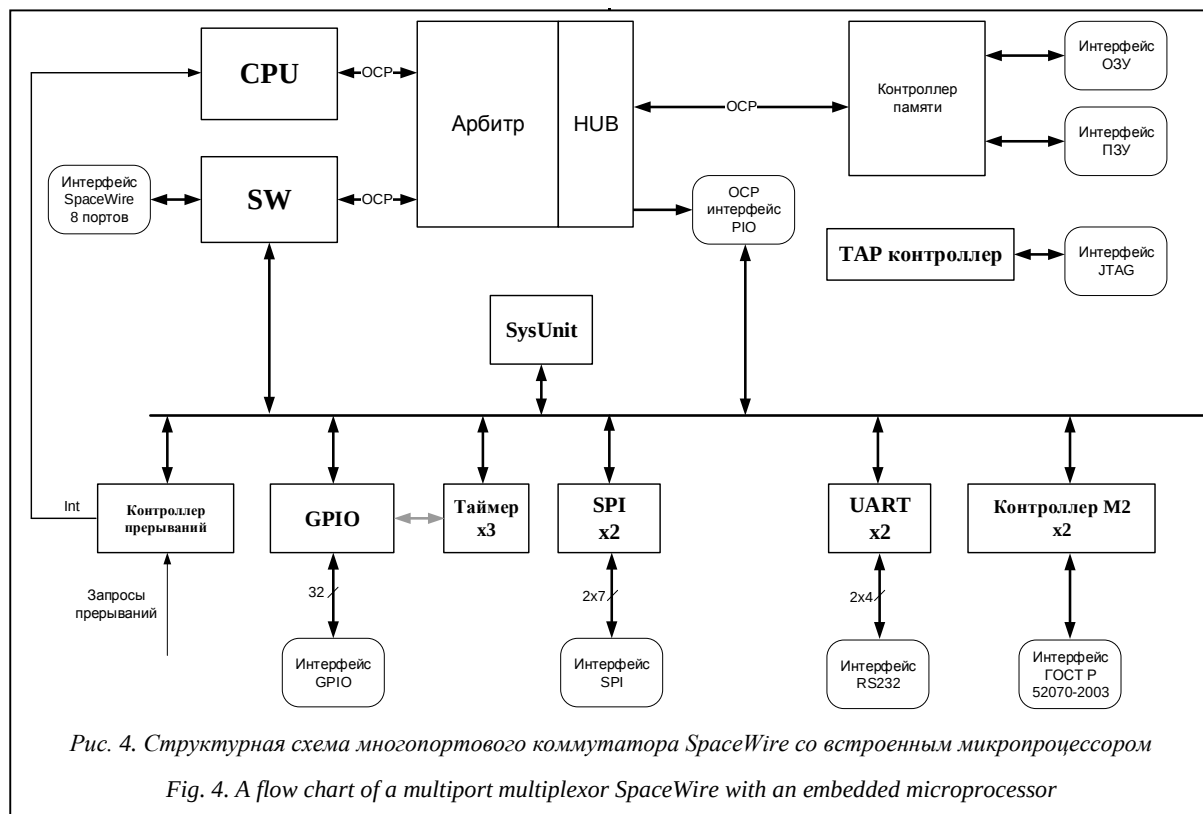
гностических регистров доступно через интерфейс I2C, либо отдельные интегрированные сигналы сбоев и ошибок могут быть выведены на порт дискретных сигналов для дальнейшего анализа системой контроля исправности (СКИ), которая в данном случае упрощается и соответственно повышается надежность.

Микросхема многопортового коммутатора SpaceWire со встроенным микропроцессором (ОКР «Схема-23») проектируется в НИИСИ РАН, срок окончания ОКР – 2016 г. Технология изготовления – КНИ 0,24 мкм. Структурная схема коммутатора представлена на рисунке 4.

Тактовая частота микросхемы – 100 МГц. Максимальная потребляемая мощность – не более 6 Вт. Предусмотрены режимы отключения неиспользуемых функциональных блоков с целью понижения потребляемой мощности.

При выполнении функций коммутатора для работоспособности микросхемы требуется малое количество элементов, например, не требуются микросхемы внешнего ОЗУ.

Для определения возможности применения рассмотренных микросхем в космических аппаратах был проведен комплекс исследований. Получены следующие характеристики: пороговая величина линейных потерь энергии *тяжелых заряженных частиц* (ТЗЧ) по тиристорному эффекту – не менее 80 МэВ×см²/мг (эффект не наблюдался); пороговая величина линейных потерь энергии ТЗЧ по одиночным событиям – не менее 6 МэВ×см²/мг; сечение насыщения по одиночным событиям при воздействии ТЗЧ – не более 3×10⁻⁸ см²/бит; уровень стой-



кости к накопленной дозе – 500 крад. Высокие параметры по стойкости объясняются прежде всего созданием в НИИСИ РАН специального технологического процесса производства микросхем с нормами 0,25–0,35 КНИ, а также разработкой радиационно стойкой библиотеки. Диапазон температур функционирования микросхем – от –60 до +125 °С. Таким образом, эти микросхемы позволяют решить большую часть задач создания вычислительных систем космического применения.

Для определения возможности использования микросхем в высокотемпературных приложениях были проведены исследования микросхем при температуре функционирования 150 °С. Получены следующие результаты. При данной температуре частота функционирования в среднем падает на 10 %, рост потребляемой мощности незначителен, что говорит о запасе в проекте и технологии изготовления. Длительность функционирования в таком режиме не определялась. Для создания полноценного процесса изготовления высокотемпературных микросхем необходима модернизация технологического процесса: изменение типа примеси, переход на вольфрамовые или комбинированные (алюминий с вольфрамом) со специальной конструкцией проводники и их утолщение и пр. Тем не менее, показана принципиальная возможность создания высокотемпературных микросхем космического применения на базе имеющихся проектов.

В результате проделанной работы достигнута унификация микросхем космического применения

на уровне ядра, внешних интерфейсов и среды обмена данными. Для сокращения потребляемой мощности и повышения надежности среда передачи данных сделана иерархической. На верхнем уровне находится коммутируемая среда RapidIO со скоростью обмена до 1 Гбайт/с на канал. На следующем уровне – коммутируемая среда SpaceWire со скоростью обмена до 200 Мбит/с, на нижнем уровне – низкоскоростные мультиплексный канал по ГОСТ Р 52070-2003 и RS232/RS485. Интерфейс Ethernet является технологическим и предусмотрен также для связи с ранее выпускаемыми модулями. Коммутируемые среды позволяют создавать различные архитектуры системы с резервированием в рамках стандартных решений для таких сред.

Температура функционирования микросхем от –60 до +125 °С. Проведенные исследования показали принципиальную возможность модернизации технологического процесса изготовления микросхем и имеющихся проектов для достижения температур функционирования порядка +200 °С.

Литература

1. Ватсон Дж., Кастро Г. Особенности применения электронных компонентов при температурах свыше 150 °С // Электронные компоненты и системы. 2012. № 7. С. 3–10.
2. Texas Instruments. URL: <http://www.ti.com/product/> (дата обращения: 25.07.2015).
3. Drivers/High-Temperature ICs. URL: <http://www.atmel.com/products/automotive/drivers/> (дата обращения: 25.07.2015).
4. CISOID. URL: <http://www.cisoid.com/> (дата обращения: 25.07.2015).
5. Кульчицкий В.В. Интеллектуализация скважинных си-

стем – инновационное направление разработки нефтяных и газовых месторождений. 2011. № 11. С. 27–30.

6. Черемисин А.Н., Рязанцев А.Э., Торопецкий К.В. Внутрискважинный мониторинг в концепции «умной» скважины // Вестн. ЦКР Роснедра. 2014. № 1. С. 2–6.

7. Shepherd P., Smith S.C., Holmes J., Francis A. M., Chiolino N., Mantooth H.A. A robust, wide-temperature data transmission system for space environments. *Aerospace Conf., IEEE* 2–9, March 2013, pp. 1–13.

8. Willet M., Ward M., Behbahani A. Using high temperature

electronics for distributed controls of turbine engines. 46th AIAA/ASME/SAE/ASEE Join Propulsion Conf. \$ Exhibit, 25–28, July 2010, Nashville, TN, pp. 1–13.

9. Бобков С.Г. Импортзамещение элементной базы вычислительных систем // Вестн. РАН. 2014. Т. 84. № 11. С. 1010–1016.

10. Бобков С.Г., Еремин А.В., Кондратьева Н.В., Сердин О.В. Разработка высоконадежных многопроцессорных модулей на базе высокоскоростных каналов RapidIO // Программные продукты и системы. 2013. № 4. С. 49–55.

DOI: 10.15827/0236-235X.112.062-069

Received 27.07.15

THE PROBLEM OF CREATING HIGH TEMPERATURE COMPUTING SYSTEMS

*(The work has been done with support of RFBR grant no. 14-29-09207 ofi_m
“Basic aspects of constructive and circuit engineering arrangement of the elements
in radiation-resistant data storage devices for high-temperature microelectronics”)*

Bobkov S.G., Dr.Sc. (Engineering), Professor, Deputy Director, bobkov@cs.niisi.ras.ru

*(Federal State Institution "Scientific Research Institute for System Analysis of the Russian Academy of Sciences"
(SRISA RAS), Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation;*

*National Research Nuclear University «MEPhI»,
Kashirskoe Highway 31, Moscow, 115409, Russian Federation)*

Abstract. The article considers the problems of creating high-temperature electronic components for oil, gas, aviation and space industries. It also describes the ways to develop Intelligent Wells (IW) systems. Creation of domestic Intelligent Wells systems would solve not only an import substitution problem; it would also allow reducing the equipment cost significantly. Thus, the cost of electronically controlled valves near perforations of an Intelligent Wells could be reduced from several hundreds of thousands dollars to several tens of thousands. The space industry has similar problems. The price of Western space-related computer can reach half a million euros, and our domestic analogue can be 10 times cheaper. In addition to high radiation resistance, flights to deep space will require functioning in a wide temperature range. The paper considers basic SRISA chips for space applications with functioning temperature up to +125°C, as well as the ways to increase the temperature range. The paper also shows the possibility of creating high temperature chips based on them.

To determine the possibility of using these chips in high temperature applications, there have been conducted some studies of the chips with the operation temperature 150°C. The results showed that at this temperature the operation frequency falls by 10% on average, increasing of power consumption is insignificant. This indicates that the stocks in the project and manufacturing technology are enough.

Keywords: high temperature electronics, Intelligent Well, SOI technology, system on chip, SpaceWire switch, Serial RapidIO switch, microprocessor.

References

1. Watson J., Castro G. High-Temperature Electronics Pose Design and Reliability Challenges. *Elektronnye komponenty i sistemy* [Electronic Components and Systems]. 2012, no. 7, pp. 3–10 (in Russ.).
2. Texas Instruments. Available at: <http://www.ti.com/product/> (accessed July 25, 2015).
3. Drivers/High-Temperature ICs. Available at: <http://www.atmel.com/products/automotive/drivers/> (accessed July 25, 2015).
4. CISSOID. Available at: <http://www.cisoid.com/> (accessed July 25, 2015).
5. Kulchitsky V.V. *Intellektualizatsiya skvazhinnykh sistem – innovatsionnoe napravlenie razrabotki neftyanykh i gazovykh mestorozhdeniy* [Intellectualization of Bore-hole Systems – Innovative Development of Oil and Gas Fields]. 2011, no. 11, pp. 27–30 (in Russ.).
6. Cheremisin A.N., Ryazantsev A.E., Toropetsky K.V. Borehole monitoring in the concept of an Intelligent Well. *Vestn. TsKR Rosnedra* [Bulletin of TsKR Rosnedra]. 2014, no. 1, pp. 2–6 (in Russ.).
7. Shepherd P., Smith S.C., Holmes J., Francis A.M., Chiolino N., Mantooth H.A. A robust, wide-temperature data transmission system for space environments. *Aerospace Conf., IEEE*. 2013, pp. 1–13.
8. Willet M., Ward M., Behbahani A. Using high temperature electronics for distributed controls of turbine engines. *Proc. of the 46th AIAA/ASME/SAE/ASEE Join Propulsion Conf. \$ Exhibit*. Nashville, TN, 2010, pp. 1–13.
9. Bobkov S.G. Import substitution of computer systems element base. *Vestn. RAN* [Bulletin of RAS]. 2014, vol. 84, no. 11, pp. 1010–1016.
10. Bobkov S.G., Yeremin A.V., Kondrateva N.V., Serdin O.V. Design of high reliability multiprocessor modules based on high-performance RapidIO interconnect architecture. *Programmnye Produkty i Sistemy* [Software & Systems]. 2013, no. 4, pp. 49–55.

УДК 004.921

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.070-074

МОДЕЛИРОВАНИЕ ПОДСТИЛАЮЩЕЙ ПОВЕРХНОСТИ В ИМИТАЦИОННЫХ СИСТЕМАХ

*(Работа выполнена при поддержке РФФИ, грант № 14-07-00020-а)**К.А. Мамросенко, к.т.н., зав. отделом, kirillam@ya.ru;**В.Н. Решетников, д.ф.-м.н., профессор, руководитель**(Центр визуализации и спутниковых информационных технологий ФНЦ НИИСИ РАН,
Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

В работе приводятся методы визуализации земной поверхности и наземных объектов с описанием применения в авиационно-космической отрасли. Показаны особенности моделирования подстилающей поверхности с использованием спутниковых снимков, данных SRTM. Рассмотрена организация хранения исходных данных в форматах Tile Map Service и Keyhole Markup Language. С развитием систем дистанционного зондирования Земли, а также систем обработки полученных данных стали применяться методы визуализации земной поверхности с использованием спутниковых снимков. Например, со спутника GeoEye-1 можно получить снимки с разрешением 0,5 метра на пиксель. В целях оптимизации производительности такие снимки хранят в виде тайлов – небольших изображений одинакового размера, которые являются фрагментами большого изображения.

Описан способ повышения производительности подсистемы визуализации при отображении множественных однотипных объектов, при котором выполняется последовательный обход всех узлов модели для клонирования и на основе данных собирается новая модель, содержащая необходимое количество экземпляров исходной модели. Массив вершин исходной модели изменяется с целью помещения экземпляра модели в необходимое пространственное положение. Массивы нормалей и текстурных координат копируются без изменений. Однако возможны ситуации, при которых число вершин в модели больше числа текстурных координат, в этом случае для объединения геометрий на место недостающих текстурных координат записываются значения из следующего экземпляра модели, что может привести к неправильному наложению текстуры. Во избежание подобной ситуации в недостающие текстурные координаты записываются нулевые значения.

Ключевые слова: тренажерно-обучающие системы, визуализация, ландшафт, TMS.

Для воссоздания ощущений реального мира при проведении тренировочных занятий необходимо воспроизвести все сигналы, получаемые оператором во время выполнения операций. Если точно воспроизвести сигналы невозможно, используются их упрощенные модели. Для ряда тренажеров некоторые группы сигналов могут не воспроизводиться из-за невостребованности в данной тренировке, что приведет к снижению сложности тренажера и его удешевлению. Сигналы от тренажера должны быть согласованными, так как мозг человека для получения полной картины происходящего совмещает сигналы, получаемые различными органами чувств [1].

В гражданской авиации наличие функционала моделирования аэропортов и земной поверхности согласно руководству ICAO Doc 9625, vol. I, 2009 является одним из требований к тренажерным системам [2, 3]. Детализация воспроизведения окружающей обстановки напрямую связана с производительностью тренажера. Для повышения эффективности использования ресурсов тренажера необходимо определить, какие группы объектов наиболее важны для достижения целей тренировки, и, следовательно, детально воспроизводить именно эти группы.

С развитием систем дистанционного зондирования Земли, а также систем обработки полученных данных визуализация земной поверхности возможна с использованием спутниковых снимков. Например, со спутника GeoEye-1 можно полу-

чать снимки с разрешением 0,5 метра на пиксель. В целях оптимизации производительности снимки хранятся в виде тайлов – небольших изображений одинакового размера, которые являются фрагментами большого изображения.

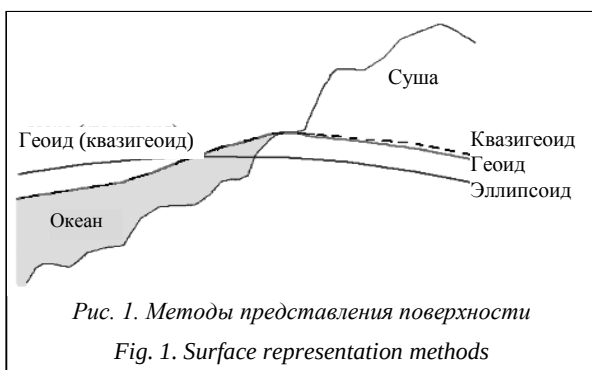
Необходимо отметить, что на снимках будут видны объекты, находящиеся на поверхности Земли: деревья, дома, тени от домов, самолеты и другие. Для некоторых участков необходима очистка изображений от подобных объектов, такая работа, как правило, выполняется вручную и весьма трудоемка.

Спутниковые снимки должны иметь пространственную привязку – набор параметров, определяющих систему координат и другие пространственные свойства данных. Данные изображения и пространственная привязка могут храниться в различных форматах – в виде GeoTIFF-файлов, в виде jpg-файла и файла с данными привязки dat и пр. Возможно преобразование формата хранения спутниковых данных, такое преобразование необходимо и в случае, когда на вход подсистемы построения виртуальной земной поверхности необходимо подать файлы в формате GeoTIFF.

Для выполнения преобразований можно воспользоваться утилитой gdal_translate. Зная координаты краев изображения l_1-1.jpg, которые хранятся в dat-файле, можно получить GeoTIFF-файл: `gdal_translate.exe -of GTiff -a_srs "+proj=latlong +ellps=WGS84 +datum=WGS84" -a_ullr 55.4097175598145 51.8169457013069 55.4151141643524`

51.8135349076488 1_1-1.jpg A1_1-1.tif. Полученный GeoTIFF-файл будет содержать изображение, а также привязку с параметрами: геодезическая система координат, эллипсоид WGS84, датум WGS84.

Геоид – фигура Земли, ограниченная поверхностью, отвесные линии к которой всюду перпендикулярны и которая проходит через точку начала отсчета высот, закрепленную на высоте среднего уровня моря (рис. 1). Эта поверхность близка к уровням морей и океанов в состоянии покоя и равновесия [4].



WGS84 – Всемирная геодезическая система координат. Поверхность, которой аппроксимируют фигуру всей Земли (геоида), – геоцентрический (глобальный) эллипсоид вращения WGS84 [5].

Для более точной аппроксимации на некотором участке земной поверхности используются референц-эллипсоиды, например, территории отдельной страны или нескольких стран. В России используются эллипсоид Красовского и Система геодезических координат СК-95. Геодезические даты, датум (Datum) – линейные и угловые смещения референц-эллипсоидов относительно центра масс Земли [6].

Для формирования рельефа подстилающей поверхности возможно использование данных, полученных с помощью радарной топографической съемки Shuttle radar topographic mission (SRTM). Для Евразии разрешение данных – 90 метров [7]. Нет SRTM-данных для самых северных (>60) и самых южных (>54) широт.

Проведено сравнение данных SRTM версии 3 с данными аэромагнитной съемки со среднеквадратической погрешностью 0,92 м. Максимальные расхождения между данными достигают по высоте 30 м, причем имеют разный знак. Расхождения обнаружены в основном в областях с переменным рельефом (долины рек), что можно объяснить неточным определением плановых координат точек в методе SRTM [8].

Также необходимо отметить, что при наличии растительности и строений данные высоты измеряются с учетом этих объектов. Следовательно, при проектировании зон повышенного интереса, например аэродромов, необходимо получить дополнительные топографические данные.

Данные SRTM могут быть получены в формате GeoTIFF. Данные являются 16-битным растром (65 536 значений, без заголовка), а значение пикселя – высотой над уровнем моря в данной точке, оно также может принимать значение –32 768, что означает «нет данных» для некоторых участков. Горизонтальный датум – WGS84. Для вычисления высот используется модель геоида EGM96 (Earth Gravitational Model 1996) [9].

При моделировании ландшафта с использованием данных SRTM возможно некорректное отображение местности. Например, прибрежный обрыв может быть смоделирован как пологий берег. Такое возможно, например, в случае недостаточного разрешения исходных данных, что приводит к интерполяции данных.

Полученные исходные данные необходимо оптимизировать для их отображения в системе визуализации. Данные спутниковых снимков для их визуализации должны быть помещены в память видеокарты. Для протяженных участков земной поверхности объем требуемой памяти значителен. Также требуются значительные вычислительные мощности для обработки данных. Имеются ограничения на максимальное разрешение текстуры. Например, изображение участка в 18 км² имеет разрешение 60 000×30 000 пикселей.

Решением может быть разбиение одного большого изображения на несколько изображений – тайлов, например размером 256×256 пикселей, которые можно быстро обработать. В случае отображения протяженных участков возрастает удаленность наблюдателя от поверхности с текстурой, следовательно, нет необходимости использовать исходные данные самого высокого разрешения. Можно сформировать несколько вариантов снимков одного и того же участка земной поверхности с различным разрешением для каждого диапазона удаления наблюдателя от виртуальной поверхности Земли. Указанные методы применяются в форматах хранения геодезических данных, таких как Tile Map Service (TMS). Подобные преобразования можно проводить одновременно с визуализацией в случае, если производительность не является одной из основных характеристик. В тренажерных системах такой подход неприемлем ввиду ограничений на время генерации одного кадра – максимум 40 мс. Следовательно, необходимо подготовить данные заранее. Все вышеизложенное относится и к данным рельефа подстилающей поверхности в случае их хранения в растровом формате типа GeoTIFF.

Для создания TMS-хранилища из исходных данных возможно использование утилиты `osgearth_package`. Пример использования: `osgearth_package config.earth --tms --max-level 19 --out d:\tms\`. Параметр `--max-level` необходимо устанавливать в зависимости от разрешения исходных данных. При недостаточном разрешении исходных

данных будет произведена экстраполяция данных изображения, что не приведет к улучшению качества итогового изображения.

Пример скрипта для генерации TMS-хранилища:

```
<map type="geocentric" version="2">
<! Загрузка слоя с данными изображения Земли>
<image name="MyMap" driver="gdal">
  <url>g:\testAir\gtif</url>
</image>
<! Загрузка слоя с данными ландшафта Земли>
<elevation name="srtm" driver="gdal">
  <url>g:\testAir\srtm</url>
  <extensions>tif</extensions>
</elevation>
</map>
```

Пример файла для визуализации данных из TMS-хранилища:

```
<map type="geocentric" version="2">
  <image name="ImMap" driver="tms">
    <url>G:\testAir\fs_cache\MyMap\tms.xml</url>
  </image>
  <elevation name="ElMap" driver="tms">
    <url>g:\testAir\fs_cache\srtm\tms.xml</url>
  </elevation>
</map>
```

Для хранения информации об объектах на поверхности Земли (аэродромах, самолетах) можно использовать файл формата KML (Keyhole Markup Language) на основе XML. Спецификация KML подразумевает, что сами модели хранятся в файлах формата Collada. Целесообразно преобразовать все модели в бинарный формат для сокращения времени загрузки сцены. При преобразовании моделей в формат Collada возможно появление графических артефактов, например черных тонких линий, разделяющих составные элементы модели. Учитывая, что число моделей может быть большим, потребуется автоматическое преобразование данных.

На одном участке поверхности могут располагаться несколько однотипных объектов, отличаю-

щихся только расположением. При загрузке каждого объекта по отдельности создается новый элемент сцены, имеющий свои данные о геометрии и текстуре. Первая проблема связана с заполнением памяти видеокарты несколькими копиями одинаковых текстур. Решением может быть использование ссылки на уже загруженную текстуру в памяти. Для реализации необходимо пройти по узлам графа сцены с целью найти узел с данной моделью.

Вторая проблема – дублирование данных о геометрии объектов.

Данные о сцене можно хранить в структуре, называемой графом сцены. Граф сцены – это иерархическая древовидная структура данных, которая организует данные в пространстве для эффективного отображения [10]. Узлы могут быть нескольких типов (рис. 2).

Узел типа Geode (Geometry Node) (рис. 3) предназначен для хранения объектов Drawables [11]. Drawables не являются узлами сцены, они присоединяются к узлам Geode, позволяют использовать в сцене данные об отображаемых объектах. Объект Geometry является одним из типов отображаемых объектов, хранит массивы вершин, текстурные координаты, данные о цвете и массивы нормалей.

Узел типа Group содержит информацию о дочерних узлах. Узел Transform предназначен для трансформации дочерних узлов, например, для изменения положения объекта при помощи матриц преобразования MatrixTransform (рис. 4) [11].

Пример загрузки 2 000 одинаковых моделей в сцену приведен на рисунке (<http://www.swsys.ru/uploaded/image/2015-4-dop/11.jpg>). Создаются узел типа Group и 2 000 дочерних узлов Transform с объектами MatrixTransform. Загружается модель: 34 узла Geode, 34 элемента Geometry. Каждый узел Transform использует копии моделей, следовательно, копии узлов Geode, объектов Geometry и объектов Drawables. В сцене хранится 68 000 Geode, 68 000 Geometry, 68 000 Drawables.

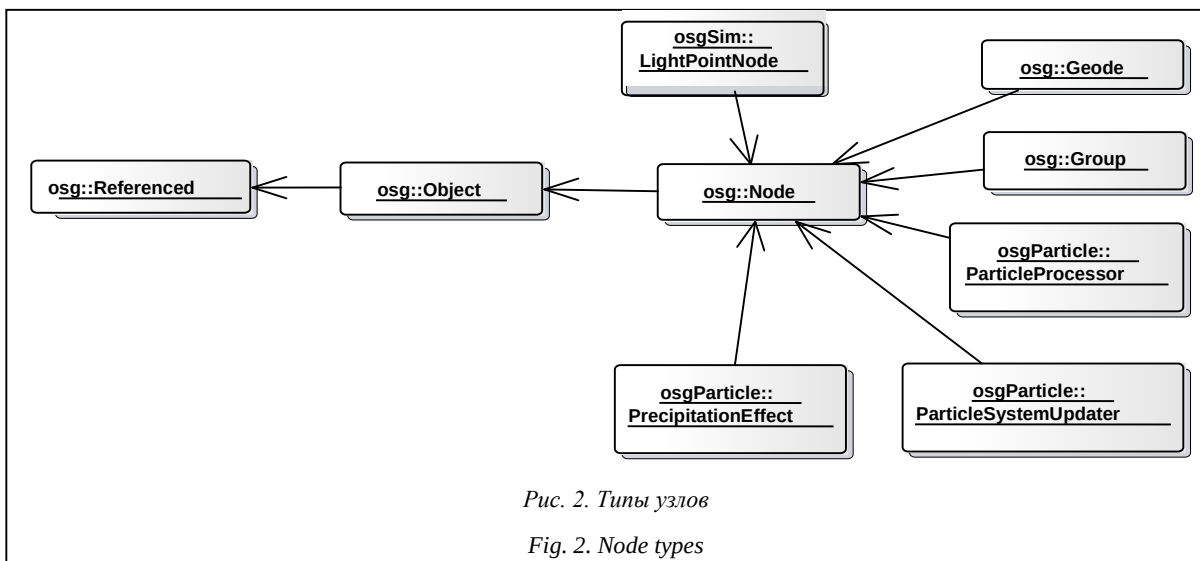
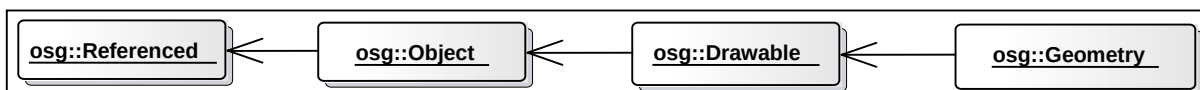
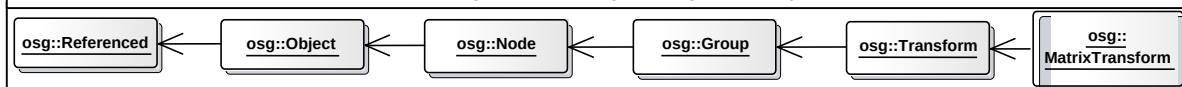


Рис. 2. Типы узлов

Fig. 2. Node types

Рис. 3. Диаграмма классов *osg::Geometry*Fig. 3. Class diagram *osg::Geometry*Рис. 4. Диаграмма классов *osg::MatrixTransform*Fig. 4. Class diagram *osg::MatrixTransform*

Одной из причин низкой производительности является то, что каждый объект *Drawable* создает свой *Display List* – поименованное множество команд OpenGL. Обработка одного *Display List* занимает фиксированное время вне зависимости от объема выполняемой работы. С уменьшением числа *Geometry* уменьшается и число *Drawables*. Уменьшение числа *Geometry* возможно за счет объединения *Geometry*. Аналогичные проблемы с производительностью имеются при использовании *vertex buffer object* OpenGL. В методе предусмотрено создание буферов для вершин и массива атрибутов вершин каждого *Geometry*. Объединение *Geometry* возможно, если все объединяемые *Geometry* используют одинаковую текстуру (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/12.jpg>) [12].

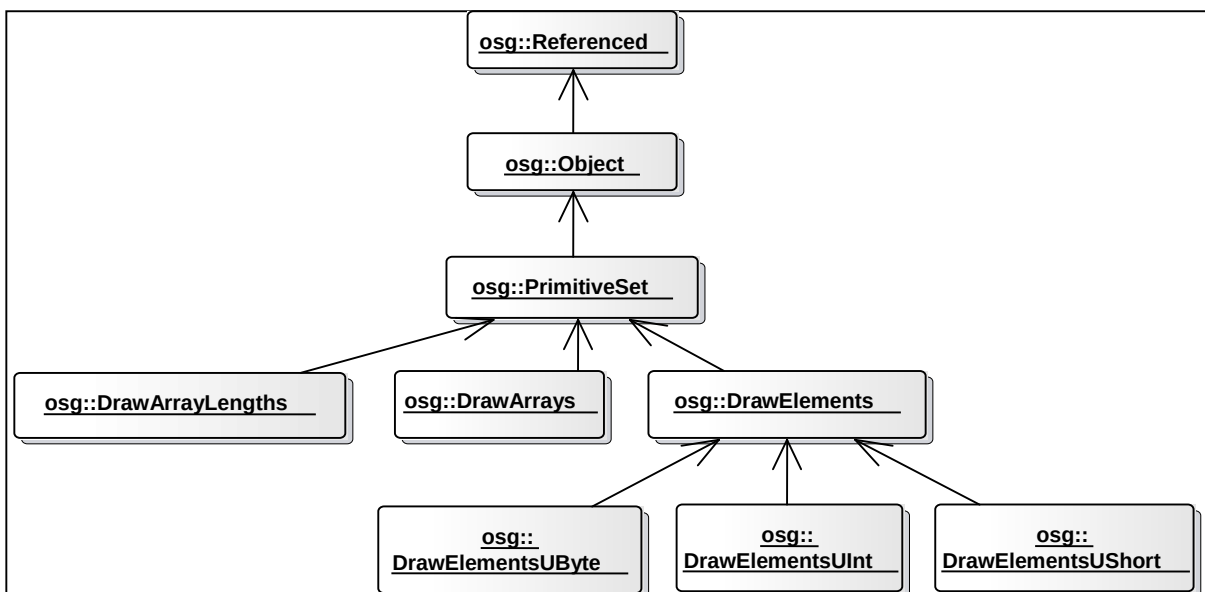
Алгоритм выполняет последовательный обход всех узлов модели для клонирования и, используя данные, собирает новую модель, содержащую необходимое количество экземпляров исходной модели. Массив вершин (*VertexArray*) исходной модели изменяется с целью помещения экземпляра модели в необходимое пространственное положение. Массивы нормалей (*NormalArray*) и текстур-

ных координат (*TexCoordArray*) копируются без изменений. Однако возможны ситуации, при которых число вершин в модели больше числа текстурных координат. В случае объединения *Geometry* на место недостающих текстурных координат будут записаны значения из следующего экземпляра модели, что приведет к неправильному наложению текстуры. Во избежание подобной ситуации в недостающие текстурные координаты записывается значение (0,0).

Объект *PrimitiveSet* предоставляет высокоуровневую поддержку команд OpenGL для отрисовки массива вершин. Он используется для определения типов примитивов, которые будут отрисовываться на основании данных, хранимых в *Geometry* (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/13.jpg>).

Из *PrimitiveSet* копируются данные об индексах. Для экземпляров модели к текущему значению индекса прибавляется наибольшее значение индекса у предыдущего экземпляра. Дочерний класс от *PrimitiveSet* *DrawElementsUInt* (рис. 5) инкапсулирует OpenGL функцию *glDrawElements()*.

В заключение отметим, что использование методов визуализации и аппаратных платформ позво-

Рис. 5. Диаграмма классов *osg::DrawElementsUInt*Fig. 5. Class diagram *osg::DrawElementsUInt*

ляет создавать высокореалистичное отображение земной поверхности в тренажерно-обучающих системах. Хранение исходных данных блоками в формате Tile Map Service позволило отображать подстилающую поверхность реальных протяженных ландшафтов с приемлемой производительностью. В работе рассматривалась виртуальная сцена с 2 000 однотипных трехмерных моделей. Всего в сцене содержится 6 млн вершин, частота обновления порядка 12 кадров в секунду. После внедрения разработанного метода частота кадров выросла до 83, ранее частота кадров не отвечала требованиям к ТОС: минимально – 25 кадров в секунду.

Необходимо отметить, что при аэрофотосъемке будут видны объекты, находящиеся на поверхности Земли: деревья, дома, самолеты и другие, а также тени от них. Для некоторых участков будет необходима очистка изображений от подобных объектов. Данная работа, как правило, выполняется вручную и весьма трудоемка.

Литература

1. Allerton D. Principles of flight simulation. Chichester, UK, Wiley, 2009, 471 p.

2. Решетников В.Н. Космические телекоммуникации. Тригонометрические воспоминания. СПб: Изд-во РИП СПб, 2015. 138 с.

3. Гиацинтов А.М. Отображение разнородных видеоматериалов на гранях трехмерных объектов в подсистеме визуализации тренажерных обучающих систем // Программные продукты и системы. 2012. № 3. С. 80–86.

4. Геоинформационный портал ГИС-Ассоциации – Геоид. URL: <http://www.gisa.ru/13065.html> (дата обращения: 28.04.2015).

5. Решетников В.Н. Космические телекоммуникации (начала). М.: Изд-во МАТИ, 2013. 184 с.

6. Системы координат и проекции. URL: <http://www.gis-craft.ru/info/cs/cs.shtml> (дата обращения: 28.04.2015).

7. GIS-Lab: Описание и получение данных SRTM. URL: <http://gis-lab.info/qa/srtm.html> (дата обращения: 28.04.2015).

8. Высотные данные SRTM против топографической съемки: все о геологии (geo.web.ru). URL: <http://geo.web.ru/db/msg.html?mid=1177761> (дата обращения: 28.04.2015).

9. GIAR-CSI SRTM 90m DEM Digital Elevation Database. URL: <http://srtm.csi.cgiar.org/SRTMdataProcessingMethodology.asp> (дата обращения: 28.04.2015).

10. Martz P. OpenSceneGraph Quick Start Guide. California: Computer Graphics Systems Development Corp., 2007, 122 p.

11. OpenSceneGraph: Class List. URL: <http://trac.openscenegraph.org/documentation/OpenSceneGraphReferenceDocs/annotated.html> (дата обращения: 28.04.2015).

12. Wang R., Qian X. OpenSceneGraph 3 cookbook. Birmingham, UK, Packt Publ., 2012, 427 p.

DOI: 10.15827/0236-235X.112.070-074

Received 27.07.15

TERRAIN MODELLING IN SIMULATION TRAINING SYSTEMS

(The work has been done with support of RFBR, grant no. 14-07-00020-a)

Mamrosenko K.A., Ph.D. (Engineering), Head of Department, kirillam@ya.ru SIT;

Reshetnikov V.N., Dr.Sc. (Physics and Mathematics), Professor, Supervisor

(Center of Visualization and Satellite Information Technologies SRISA, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation)

Abstract. The work presents the methods of the earth's surface and ground elements visualization and their usage in the aerospace industry. It also shows the features of the underlying surface modeling using satellite imagery, SRTM data. The authors consider how initial data is stored in the Tile Map Service and Keyhole Markup Language formats. Development of the earth remote sensing and data processing systems allowed using earth surface visualization techniques with satellite images as textures. For example, the GEOEYE-1 satellite makes it possible to obtain images with the resolution of 0.5 meters per pixel. In order to optimize performance the images are stored as tiles, which are small images of the same size, fragments of a big picture. The paper describes the method for increasing the visualization subsystem performance when displaying multiple objects of the same type. This method makes the sequent bypassing of all nodes of the model to clone and assembles a new model, which contains the required number of copies of the original model. An array of vertices of the original model is changed in order to place a sample model in the desired spatial position. An array of normals is copied without changes, as well as an array of texture coordinates. However, it is possible that the number of vertices in the model is bigger than the number of texture coordinates. In this case missing texture coordinates are replaced by the values from the next sample model, which can lead to incorrect texture mapping. To avoid such situations missing texture coordinates values are set to zero.

Keywords: training simulation systems, rendering, landscape, TMS.

References

1. Allerton D. Principles of flight simulation. Chichester UK, Wiley Publ., 2009, 471 p.
2. Reshetnikov V.N. Kosmicheskie telekommunikatsii. Trigonometricheskie vospominaniya [Space telecommunications. Trigonometric memories]. St. Petersburg, RIP SPB Publ., 2015, 138 p.
3. Giatsintov A.M. Playback of heterogeneous videos on 3d object's edges in visualization subsystem of training simulation system. *Programmnye produkty i sistemy* [Software & Systems]. 2012, no. 3, pp. 80–86 (in Russ.).
4. *Geoinformatsionny portal GIS-Assotsiatsii – Geoid* [Geoinformation Portal of GIS-Association – Geoid]. Available at: <http://www.gisa.ru/13065.html> (accessed April 28, 2015).
5. Reshetnikov V.N. Kosmicheskie telekommunikatsii (nachala) [Space Telecommunications (basics)]. Moscow, ITTS MATI Publ., 2013, 2nd ed., 184 p.
6. *Sistemy koordinat i proyeksii* [Frames of Axis and Projections]. Available at: <http://www.gis-craft.ru/info/cs/cs.shtml> (accessed April 28, 2015).
7. *GIS-Lab: Opisanie i poluchenie dannykh SRTM* [GIS-Lab: a Description and Obtaining SRTM Data]. Available at: <http://gis-lab.info/qa/srtm.html> (accessed April 28, 2015).
8. *Vysotnye dannye SRTM protiv topograficheskoy syemki – vse o geologii (geo.web.ru)* [SRTM Upper-air Data versus Mapping: All about Geology]. Available at: <http://geo.web.ru/db/msg.html?mid=1177761> (accessed April 28, 2015).
9. *CGIAR-CSI SRTM 90m DEM Digital Elevation Database*. Available at: <http://srtm.csi.cgiar.org/SRTMdataProcessingMethodology.asp> (accessed April 28, 2015).
10. Martz P. OpenSceneGraph Quick Start Guide. California: Computer Graphics Systems Development Corp., 2007, 122 p.
11. OpenSceneGraph: Class List. Available at: <http://trac.openscenegraph.org/documentation/OpenSceneGraphReferenceDocs/annotated.html> (accessed April 28, 2015).
12. Wang R., Qian X. OpenSceneGraph 3 Cookbook. Birmingham, UK, Packt Publ., 2012, 427 p.

УДК 004.522

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.075-078

ЭРГОНОМИЧНЫЙ ГОЛОСОВОЙ ИНТЕРФЕЙС УПРАВЛЕНИЯ АНТРОПОМОРФНЫМ РОБОТОМ

*(Исследования выполнены при поддержке РФФИ, грант № 15-07-04544)**М.В. Михайлюк, д.ф.-м.н., профессор, зав. отделом, mix@niisi.ras.ru;**М.А. Торгашев, к.ф.-м.н., зав. сектором, mtorg@mail.ru;**Д.В. Омельченко, ведущий программист, denvikom@gmail.com**(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Антропоморфные роботы (роботы, по своему строению похожие на человека) находят широкое применение в различных областях научной и хозяйственной деятельности. Это объясняется не только психологическими моментами, но и тем, что многие объекты, созданные человеком, приспособлены для манипулирования именно его руками. Однако эргономичное управление такими роботами является серьезной и актуальной проблемой. Стандартные компьютерные интерфейсные устройства (клавиатура, мышь, джойстик и т.д.) не обеспечивают достаточный уровень эргономики. Использование пультов управления тоже затруднительно, так как каждый управляющий элемент пульта обычно воздействует на один шарнир робота. В качестве альтернативы для управления часто используют экзоскелеты, представляющие собой жесткие шарнирные конструкции, надеваемые оператором и соответствующие строению человеческого тела. С помощью экзоскелета можно реализовать так называемый копирующий режим управления, при котором робот будет в точности повторять движения оператора. Тем не менее, использование экзоскелета часто затруднительно в силу либо его большого веса, либо невозможности работы с ним в скафандре. В данной работе предлагается голосовой интерфейс управления роботом. Идея заключается в том, что при произнесении оператором команды производится ее распознавание и в случае успеха запускается на выполнение заранее подготовленный скрипт, в соответствии с которым робот выполняет заданное действие. Апробация предложенных методов и алгоритмов показала возможность их успешного использования в качестве эргономичного интерфейса голосового управления антропоморфными роботами.

Ключевые слова: *голосовой интерфейс, антропоморфный робот, имитационно-тренажерные комплексы, виртуальное моделирование.*

Широкое применение в различных областях научной и хозяйственной деятельности находят антропоморфные роботы, по своему строению похожие на человека. Это объясняется и психологическими моментами, и тем, что многие созданные человеком объекты приспособлены для манипулирования именно его руками. Однако эргономичное управление этими роботами является серьезной проблемой. Стандартные интерфейсные устройства (клавиатура, мышь, джойстик и т.д.) не могут обеспечить достаточный уровень эргономики. Затруднительно и использование пультов управления, так как каждый их элемент обычно воздействует на один шарнир робота [1]. Такое «пошарнирное» управление не только занимает длительное время, но и является неточным и неудобным. В качестве альтернативы для управления часто используют экзоскелеты – жесткие шарнирные конструкции, надеваемые оператором и соответствующие строению человеческого тела [2]. Движение рук оператора вызывает соответствующие сгибы в шарнирах экзоскелета. Если в масштабе реального времени передавать углы поворота во всех шарнирах экзоскелета в систему управления реальным роботом, то робот будет повторять движения оператора. Мы получим так называемый копирующий режим управления. Тем не менее, использование экзоскелета часто затруднительно либо из-за его большого веса, либо из-за невозможности работы с ним в скафандре. В работах [3–6] для управления роботом предлагается использовать устройство

Кинект. Таким образом, возникает задача поиска новых видов интерфейса для эргономичного управления роботом.

В данной работе предлагается использовать устройство Кинект для голосового управления роботом в командном режиме. Идея заключается в следующем: когда оператор голосом произносит команду, устройство Кинект [7] записывает ее и передает в систему управления, которая инициирует процесс распознавания этой команды. Если команда распознана, система управления запускает на выполнение заранее подготовленный скрипт, в соответствии с которым робот выполняет заданное действие.

Система управления голосовым интерфейсом

Система управления включает функциональную схему и модуль расчета управляющего сигнала. Функциональная схема создается в специальном визуальном редакторе, включающем поле редактора и библиотеку функциональных блоков. Блоки разбиты на группы (логические, арифметические, алгебраические, дифференциальные и т.д.). Пользователь может перемещать блоки из библиотеки на поле редактора и присоединять входы одних блоков к выходам других. Многие блоки имеют индивидуальные параметры настройки, которые задаются через отдельное диалоговое окно. Например, для блока «Сумматор» через диалог

настройки можно задать количество входов. Кроме этого, сами блоки, а также их входы и выходы в редакторе имеют всплывающие подсказки, что позволяет быстро получить краткую информацию о них. Все это в целом дает возможность быстро разрабатывать, анализировать и выявлять ошибки функционирования структурной схемы системы управления.

Для рассматриваемой задачи созданы два новых функциональных блока: VoiR (Voice Recognition), в котором в качестве параметра может быть записана произвольная фраза, и Script, в котором в качестве параметра записано имя скрипта. Работа блока VoiR заключается в том, что он вызывает систему распознавания речи и получает от нее фразу, распознанную системой в текущий момент времени. Далее он сравнивает эту фразу со своей фразой-параметром и в случае совпадения выдает управляющий сигнал 1. Блок Script, получая на входе 1, запускает скрипт, имя которого записано в его параметре и в соответствии с которым робот выполняет необходимые действия. На рисунке 1 показан пример функциональной схемы, включающей или выключающей лампочку по голосовой команде.

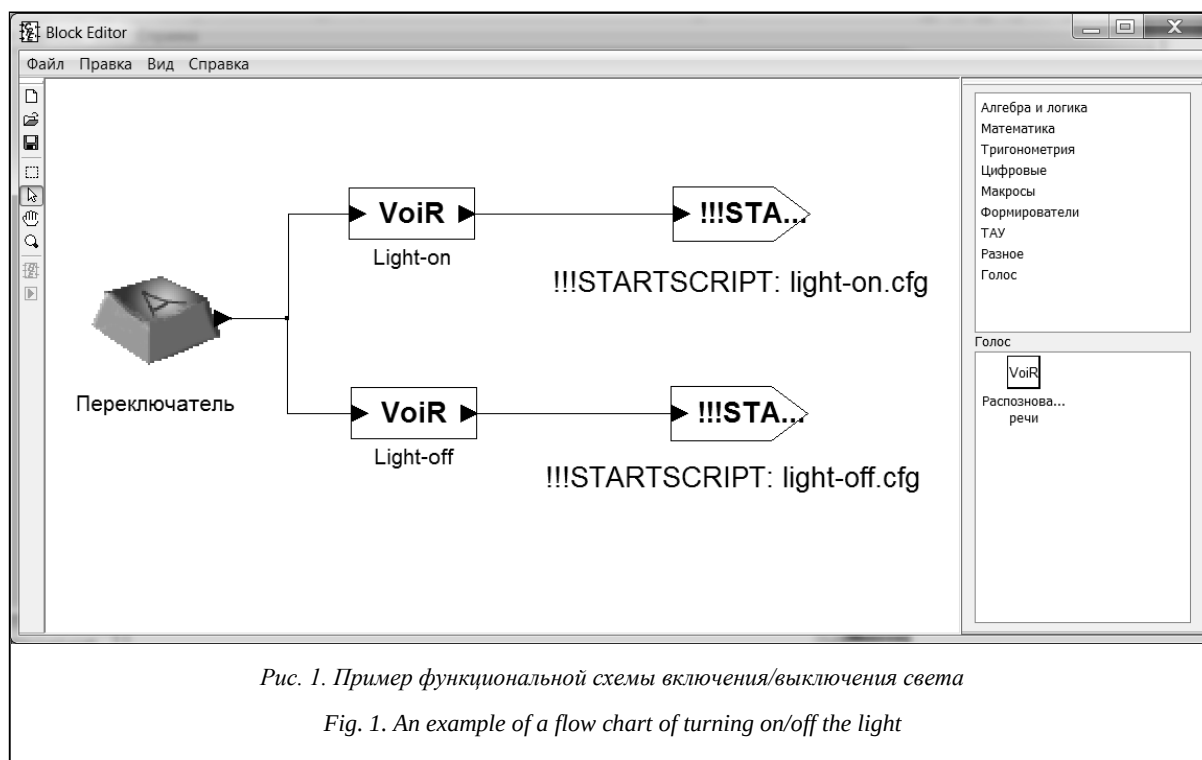
Система распознавания речи

В качестве системы распознавания речи выбрана открытая кроссплатформенная система CMU Sphinx университета Карнеги–Меллон. Она включает несколько различных словарей, в том числе русский словарь с транскрипцией, а также предоставляет возможность составить собственный сло-

варь, ориентированный на используемую семантическую область. Это и было сделано для рассматриваемой задачи. Работа системы заключается в том, что она прослушивает эфир, распознает произнесенные слова и записывает текст в буфер. Если при произнесении фразы возникает пауза, система считает, что фраза закончена, и посылает эту фразу на выход.

Система CMU Sphinx реализована несколькими компонентами [7]: Pocketsphinx – библиотека распознавания речи, написанная на языке C; Sphinxbase – базовая библиотека системы, необходимая для работы Pocketsphinx; Sphinx4 – гибкая настраиваемая библиотека распознавания речи, написанная на языке Java; Sphinxtrain – инструмент создания собственной акустической модели. Для создания приложений с функционалом распознавания речи можно использовать как библиотеку Pocketsphinx, так и Sphinx4. В данном проекте используется библиотека Pocketsphinx, так как она написана на языке C и легко интегрируется в программы, разработанные в среде Visual Studio на языке C++.

Чтобы подключить библиотеку Pocketsphinx к приложению, необходимо добавить в проект динамические библиотеки sphinxbase.dll и pocketsphinx.dll, а также файлы акустической модели, языковой модели и словаря [8, 9]. На этапе инициализации библиотеке передаются различные параметры, главными из которых являются путь к директории, содержащей файлы акустической модели (библиотека реализует скрытую марковскую акустическую модель), путь к файлу триграммной языковой модели [10] и путь к файлу словаря. Рас-



познавание звука возможно как с микрофона, так и из звукового файла. Режим распознавания также указывается в качестве параметра.

На этапе работы приложения в основном цикле программы последовательно вызываются функции записи звука с микрофона и распознавания записанных фраз. В случае успешного распознавания возвращается текстовая строка, содержащая распознанную фразу.

Система создания и выполнения скриптов

Важнейшая составляющая в идеологии управления робототехническими системами – работа со скриптовыми командами. Скриптовая команда представляет собой совокупность текстовой строки, задающей назначение и тип команды, и связанных с ней данных произвольного формата и размера. Формат команды имеет вид «**Command** = **data, data, ...**», где **Command** задает назначение и тип команды, а **data, data, ...** – набор связанных с ней данных. Такой язык дает возможность изменить практически любые параметры виртуальной сцены с помощью универсального механизма. Кроме того, эта схема позволяет организовать эффективный обмен данными произвольного типа.

Система включает большое количество поддерживаемых команд:

- системные команды – регулирование обслуживания общей очереди и поступающих блоков команд: задержка выполнения команд на заданное время, формирование отдельных блоков из команд, ожидающих наступления определенных событий, сброс и генерация событий и т.д.
- команды работы с материалами – изменение параметров материала, назначение новых текстурных карт;
- команды управления узлами сцены – изменение текущей камеры и текущего объекта для управления, управление видимостью объектов, изменение положения, ориентации и масштаба объектов;
- команды работы с анимационными треками – запуск анимации в определенном диапазоне кадров, остановка анимации, переход к некоторому кадру, создание и удаление именованных анимационных треков с привязкой к определенным узлам сцены, задание флага зацикленного воспроизведения и скорости анимации и другие;
- команды конфигурирования управления от клавиатуры, мыши и джойстика;
- команды работы со звуком;
- команды работы с выводом изображения – конфигурирование общих параметров визуализации, задание эффектов постобработки, настройка портов вывода;
- команды записи и воспроизведения лог-файлов.

В задаче голосового управления наиболее востребованными и удобными являются команды последнего типа. С их помощью можно полностью записать движение робота при выполнении операции как в случае управления с экзоскелета, так и в результате решения задачи обратной кинематики. Затем с помощью голосовой команды можно вызвать проигрывание записанной операции. Лог-файлы реализованы в подсистеме визуализации виртуальных сцен и предназначены для хранения и записи всех необходимых в симуляторе параметров – в них фиксируются положение и ориентация всех управляемых объектов, а также такие параметры, как флаги активности источников света, параметры виртуальных камер, систем частиц и других виртуальных объектов. В данной работе подготовлены лог-файлы для изменения роботом состояния интерьера космического модуля путем нажатия на соответствующие кнопки пульта управления. На рисунке 2 проиллюстрировано такое действие: включение света внутри модели модуля СО-1 осуществляется по голосовой команде оператора «Робби, включи свет».

В заключение отметим, что в статье рассмотрен разработанный в НИИСИ РАН программный пакет для реализации эргономичного интерфейса управления антропоморфным роботом с помощью голоса. Пакет включает в себя систему управления голосовым интерфейсом, систему распознавания речи и систему создания и выполнения скриптов. Для тестирования был использован имитационно-тренажерный комплекс, включающий виртуальную модель антропоморфного робота SAR-401, модель интерьера космического модуля СО-1, а также подсистемы расчета динамики и визуализации трехмерных виртуальных сцен в масштабе реального времени. Апробация предложенных методов и алгоритмов показала возможность их успешного

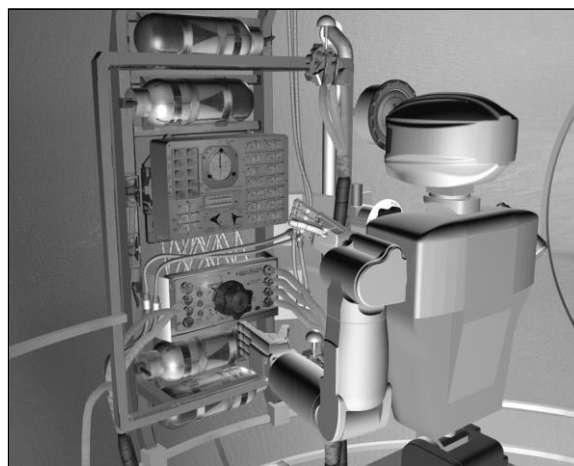


Рис. 2. Выполнение моделью робота голосовой команды оператора

Fig. 2. A robot model executes the operator's voice instruction

использования в качестве эргономичного интерфейса голосового управления антропоморфными роботами.

Литература

1. Михайлюк М.В. Виртуальные панели управления в космических тренажерах // Наука и технологии в промышленности. 2014. № 3–4. С. 36–39.
2. Бурдин Б.В., Михайлюк М.В., Сохин И.Г., Торгашев М.А. Использование виртуальных 3D-моделей для экспериментальной отработки бортовых полетных операций, выполняемых с помощью антропоморфных роботов // Робототехника и техническая кибернетика. 2013. № 1. С. 42–46.
3. Михайлюк М.В., Мальцев А.В. Реализация эргономичного интерфейса управления виртуальной моделью антропоморфного робота с использованием Kinect // Программная инженерия. 2015. № 10. С. 12–18.
4. Oikonomidis I., Kyriazis N., Argyros A. Efficient model based 3D tracking of hand articulations using Kinect. Proc. 22nd British Machine Vision Conf. 2011, pp. 101.1–101.11.

5. Ma B., Xu W., Wang S. A Robot Control System Based on Gesture Recognition Using Kinect. TELKOMNIKA Indonesian Journ. of Electrical Engineering, 2013, vol. 11, no. 5, pp. 2605–2611; URL: <http://iaesjournal.com/online/index.php/telkomnika/article/view/2493> (дата обращения: 10.07.2015).
6. Нагапетян В.Э., Толмачев И.Л. Бесконтактное управление роботизированной рукой посредством жестов человека // Вестн. РУДН. Сер. Математика. Информатика. Физика. 2014. № 2. С. 157–163.
7. Kinect for Windows SDK 1.8. URL: <https://msdn.microsoft.com/en-us/library/hh973074.aspx> (дата обращения: 10.07.2015).
8. CMUSphinx. URL: <http://cmusphinx.sourceforge.net/wiki/> (дата обращения: 10.07.2015).
9. Pocketsphinx. Распознавание речи и голосовое управление в Linux. URL: <http://habrahabr.ru/post/167479/> (дата обращения: 10.07.2015).
10. Холоденко А.Б. О построении статистических языковых моделей для систем распознавания русской речи. URL: http://www.intsys.msu.ru/invest/speech/articles/rus_lm.htm (дата обращения: 9.07.2015).

DOI: 10.15827/0236-235X.112.075-078

Received 27.07.15

ERGONOMIC VOICE INTERFACE TO CONTROL AN ANTHROPOMORPHIC ROBOT

(The research has been done with support of RFBR, grant no. 15-07-04544)

Mikhaylyuk M.V., Dr.Sc. (Physics and Mathematics), Professor, Head of Department, mix@niisi.ras.ru;

Torgashev M.A., Ph.D. (Physics and Mathematics), Head of Sector, mtorg@mail.ru;

Omelchenko D.V., Lead Programmer, denvikom@gmail.com

(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation)

Abstract. Anthropomorphic robots (robots structurally similar to human beings) are widely used in various fields of scientific and economic activity. It is due to psychological factors, as well as to the fact, that many man-made objects are adapted to manipulate by human hands. However, ergonomic control of such robots is a serious and pressing problem. Standard computer interface devices (keyboard, mouse, joystick etc.) do not provide a sufficient level of ergonomics. Using control panels is also difficult because each control panel element usually affects one joint of the robot. Or alternatively, exoskeletons are often used for control. They are rigid articulated construction worn by an operator and match the organization of human body. An exoskeleton allows implementing the so called copying control mode when the robot will follow the operator's exact movements. However, the exoskeleton is often difficult to use, either because of its great weight, or because it is impossible to use it in the suit. This paper proposes a voice interface to control a robot. The idea is that when the operator's command is pronounced it is recognized and then in case of success the prepared script starts running, so the robot performs the specified action. Testing of the proposed methods and algorithms showed that it is possible to use them as an ergonomic interface of anthropomorphic robots' voice control.

Keywords: voice interface, anthropomorphic robot, simulation and training complexes, virtual modeling.

References

1. Mikhaylyuk M.V. Virtual Control Panels in Space Simulators. *Nauka i tekhnologii v promyshlennosti* [Science and Technology in Industry]. 2014, no. 3–4, pp. 36–39 (in Russ.).
2. Burdin B.V., Mikhaylyuk M.V., Sokhin I.G., Torgashev M.A. Using Virtual 3D Models for Testing On-board Flight Operations Performed by Anthropomorphic Robots. *Robototekhnika i tekhnicheskaya kibernetika* [Robotics and Technical Cybernetics]. 2013, no. 1, pp. 42–46 (in Russ.).
3. Mikhaylyuk M.V., Maltsev A.V. Ergonomic interface Implementation to control a virtual model of anthropomorphic robot using Kinect. *Programnaya inzheneriya* [Software Engineering]. 2015, no. 10 (in Russ.).
4. Oikonomidis I., Kyriazis N., Argyros A. Efficient model based 3D tracking of hand articulations using Kinect. *Proc. of the 22nd British Machine Vision Conf.* 2011, pp. 101.1–101.11.
5. Ma B., Xu W., Wang S. A Robot Control System Based on Gesture Recognition Using Kinect. *TELKOMNIKA Indonesian Journ. of Electrical Engineering*. 2013, vol. 11, no. 5, pp. 2605–2611. Available at: <http://iaesjournal.com/online/index.php/telkomnika/article/view/2493> (accessed July 10, 2015).
6. Nagapetyan V.E., Tolmachov I.L. Contactless control of a robot arm using human gestures. *Vestnik RUDN. Seriya "Matematika. Informatika. Fizika"* [PFUR Bulletin. Ser. "Mathematics, informatics, physics"]. 2014, no. 2, pp. 157–163. (in Russ.).
7. Kinect for Windows SDK 1.8. Available at: <https://msdn.microsoft.com/en-us/library/hh973074.aspx> (accessed July 10, 2015).
8. CMUSphinx. Available at: <http://cmusphinx.sourceforge.net/wiki/> (accessed July 10, 2015).
9. Pocketsphinx. *Speech Recognition and Voice Control in Linux*. Available at: <http://habrahabr.ru/post/167479/> (accessed July 10, 2015) (in Russ.).
10. Kholodenko A.B. *On the Construction Statistical Language Models for Russian Speech Recognition Systems*. Available at: http://www.intsys.msu.ru/invest/speech/articles/rus_lm.htm (accessed July 10, 2015) (in Russ.).

УДК: 621.382 + 536.24 + 536.37
DOI: 10.15827/0236-235X.112.079-086

Дата подачи статьи: 27.07.15

КОНЦЕПЦИЯ МАТЕМАТИЧЕСКОГО И КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ТЕПЛОВЫХ ПРОЦЕССОВ В ЭЛЕКТРОННЫХ СИСТЕМАХ

*А.Г. Мадера, д.т.н., профессор, зав. отделом, alexmadera@mail.ru
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

В статье излагаются концепции теплового проектирования электронных систем, а также требования к математическому и компьютерному моделированию тепловых процессов в электронных системах. Сформулированы требования к математическим моделям, обеспечивающим адекватное моделирование тепловых процессов при тепловом проектировании электронных систем. Согласно им, математические модели должны быть динамическими, нелинейными, интервально стохастическими, учитывать влияние эффекта тепловой обратной связи (взаимодействие между тепловыми и электрическими режимами в электронной системе), а также статистический технологический разброс электрических и тепловых определяющих параметров элементов в электронной системе. Полученные уравнения математической модели тепловых процессов в электронной системе удовлетворяют сформулированным требованиям и представляют собой систему нестационарных, нелинейных, интервально стохастических дифференциальных уравнений в обыкновенных производных первого порядка. Разработанные концепции и математические модели могут быть положены в основу создаваемого многофункционального программного комплекса теплового проектирования электронных систем.

Ключевые слова: электронная система, тепловой процесс, математическая модель, компьютерное моделирование, динамический, нелинейный, интервальный, стохастический, тепловая обратная связь.

В процессе функционирования *электронной системы* (ЭС) часть ее элементов (микропроцессоры, электронные модули, высокопроизводительные вычислительные системы, электронная аппаратура, электронные приборы и др.) потребляют энергию от источников питания (микросхемы, резисторы, ряд других электрорадиоэлементов) как на полезные сигналы, так и на рассеивание в виде теплоты, приводя к нагреванию ЭС и ее элементов. Другие элементы ЭС (электрические разъемы, некоторые электрорадиоэлементы, элементы конструкции, теплоотводы) не потребляют энергию и нагреваются в результате теплообмена с теплообделяющими элементами. При этом все элементы ЭС участвуют в процессах теплообмена между собой, средой внутри и вне ЭС и системой теплоотвода и охлаждения ЭС. Таким образом, в процессе функционирования в элементах и в ЭС в целом возникают распределения температуры, или температурные поля.

Характеристики и параметры (надежность, помехоустойчивость, быстродействие, статические и динамические параметры) элементной полупроводниковой базы и ЭС различного назначения (микросхем, электронных модулей, высокопроизводительных вычислительных комплексов и систем и др.) в значительной степени зависят от температуры. Поэтому возникающие в ЭС температурные поля ввиду значительной зависимости электрических параметров элементов от температуры приводят к их существенному изменению. В свою очередь, изменения в результате нагревания электрических параметров элементов влекут за собой изменения потребляемых элементами мощностей, что вызывает изменение температурного поля. Такое влияние тепловых и электрических

режимов друг на друга называют тепловой обратной связью, или саморазогревом [1, 2].

Практика показывает, что пренебрежение тепловым воздействием на функционирование ЭС приводит к ошибкам проектирования и нежелательным последствиям (выходу ЭС из строя, ложным срабатываниям, снижению надежности, быстродействия, помехоустойчивости и др.) [3, 4]. При этом уровень температурных распределений в ЭС является зачастую решающим ограничительным фактором на пути создания конкурентоспособных ЭС. В силу этого важнейшей задачей при создании новых ЭС и элементов является разработка системы теплового проектирования, позволяющей еще на стадии проектирования ЭС получать достоверные данные о будущих параметрах и характеристиках создаваемых изделий при различных условиях испытаний и эксплуатации, а также оптимизировать конструкции элементов, ЭС, теплоотводов, систем охлаждения, обеспечивающих необходимый уровень температуры элементов, узлов и устройств ЭС.

Наблюдающаяся в настоящее время тенденция как к ужесточению требований, предъявляемых к ЭС, ее элементам и конкурентоспособности, так и к росту потребляемой микропроцессорами мощности приводит к постоянному росту актуальности теплового проектирования при создании ЭС, что обуславливает повышенные требования к уровню адекватности математического и компьютерного моделирования, являющегося фундаментом теплового проектирования ЭС.

Для создания адекватных моделей и методов моделирования тепловых режимов ЭС, обладающих высоким уровнем достоверности отражения реальности, необходимо учитывать разнообразные

сложные физические процессы, сопровождающие функционирование ЭС и ее элементов:

- нелинейный динамический и трехмерный характер тепловых процессов в ЭС [5, 6];
- динамический характер электрического режима и потребляемой мощности как элементами ЭС, так и ЭС в целом [7];
- эффект тепловой обратной связи в элементах ЭС, заключающийся во влиянии, оказываемом электрическими и тепловыми режимами элементов ЭС друг на друга; тепловая обратная связь в электронных элементах ЭС возникает в силу существенной зависимости электрических параметров и потребляемой элементами мощности от температуры, а температуры элементов – от изменения мощностей потребления элементов [1, 2];
- статистический технологический разброс электрических, тепловых и конструктивных параметров ЭС и ее элементов, который приводит к интервально стохастическому характеру тепловых процессов в ЭС [8];
- случайный характер параметров окружающей среды (температуры, скорости и направления движения среды, влажности, плотности) и хладагентов (температуры, скорости потока) в системе охлаждения ЭС [5];
- внешние воздействия различной физической природы (электромагнитные, тепловые и пр.), носящие, в принципе, нелинейный и стохастический характер, со стороны других технических систем.

Чтобы система теплового проектирования была высокоэффективным средством проектирования конкурентоспособных ЭС, а также с достаточной для практики точностью позволяла прогнозировать будущие характеристики создаваемых ЭС, необходимо наличие в ней различных функциональных возможностей: нелинейный, динамический и трехмерный характер тепловых процессов, тепловая обратная связь, интервально стохастический характер электрического и теплового режимов ЭС, стохастический характер параметров внешней и внутренней среды в ЭС.

Создать систему теплового проектирования ЭС можно только с помощью современных информационных технологий компьютерного и суперкомпьютерного моделирования, являющихся инструментом проектирования и отработки новых ЭС, а также оптимизации воздействий различной физической природы. Только располагая многофункциональными программными системами (программными пакетами/комплексами), позволяющими на этапе проектирования достоверно и адекватно прогнозировать поведение, эксплуатационные характеристики и параметры будущих ЭС, можно обеспечить конкурентоспособность новых ЭС на мировом коммерческом рынке. Решение данной проблемы особенно актуально при создании высоконадежных и помехоустойчивых ЭС и микропроцессоров, предназначенных для эксплуатации в

экстремальных условиях окружающей среды, при тепловых, механических, химических воздействиях, космических излучениях. Разработка *многофункционального программного комплекса теплового проектирования ЭС* (МФПК-ТП-ЭС) является критически важной для создания новой конкурентоспособной электронной техники.

Необходимо отметить, что известные программные комплексы для тепловых расчетов элементов и ЭС (Beta Soft, TGM, Thermal Analysis, Thermal Solution, Ansys, тепловые модули, встроенные в различные CAD), используемые в зарубежных компаниях при проектировании элементов и ЭС различного назначения, обладают рядом системных недостатков, а именно:

- методы, модели, математические и компьютерные вычислительные алгоритмы, заложенные в зарубежных САПР, как правило, недоступны пользователю, что не позволяет достоверно судить об области их применимости, адекватности, равно как и о релевантности и валидности;
- содержащиеся по умолчанию физические и конструкционные параметры и характеристики элементной базы и конструкций ЭС привязаны к зарубежным технологиям, конструкциям и стандартам, не применимым к отечественной практике создания ЭС и их элементов;
- возможности моделирования температурных полей ЭС во многом ограничены и не позволяют учитывать такие, например, важнейшие факторы ЭС, как многослойность печатных плат, многие конструктивные особенности микросхем и их корпусов, способы монтажа и пр.;
- отсутствуют возможность моделирования тепловых процессов ЭС в условиях интервально стохастической неопределенности электрических и тепловых параметров элементов и ЭС и среды как внутри, так и вне ЭС, а также учет влияния эффекта тепловой обратной связи.

Указанные недостатки существенно затрудняют использование зарубежных программных комплексов в практике теплового проектирования создаваемых отечественных электронных элементов и ЭС, что диктует необходимость создания и развития МФПК-ТП-ЭС, обладающего перечисленными выше функциональными возможностями.

Ввиду огромного разнообразия элементов и конструкций ЭС математическое и компьютерное моделирование тепловых процессов в ЭС является чрезвычайно сложной задачей. В состав ЭС входит большое количество разнородных элементов, имеющих разнообразную форму и размеры, взаимодействующих между собой, окружающей средой и хладагентами. Для реализации возможности математического и компьютерного моделирования тепловых процессов в ЭС с учетом разнообразных физических процессов, сопровождающих функционирование ЭС и ее элементов (см. выше), целесообразно использовать подход, основанный на двух

принципиальных концепциях: представление ЭС в виде иерархической структуры, отражающей различные уровни конструкции и протекающие в них процессы теплообмена [9, 10]; представление тепловой модели ЭС на каждом иерархическом уровне в виде системы N изотермических элементов, взаимодействующих между собой и средой [5, 11].

Иерархическая структура тепловых процессов в ЭС

Иерархическая структура ЭС (рис. 1) в общем случае характеризуется пятью уровнями:

1-й уровень – отдельные элементы ЭС (микросхемы и электрорадиоэлементы);

2-й уровень – электронный модуль, включающий многослойную печатную плату (МПП), установленные на ней микросхемы, электрорадиоэлементы, теплоотводы, элементы системы охлаждения, конструктивные элементы крепления и монтажа электронного модуля;

3-й уровень – панель (блок, субблок и др. – в различной терминологии), объединяющая несколько электронных модулей и содержащая конструктивные элементы их крепления и монтажа в панели, системы интерфейса, теплоотводы и систему охлаждения панели;

4-й уровень – стойка, в состав которой входят панели, элементы конструкции крепления и монтажа панелей в стойке, системы интерфейса и охлаждения стойки;

5-й уровень – помещение, где будут функционировать стойки полной ЭС (стационарное или подвижное), система интерфейса стоек, системы

охлаждения ЭС и кондиционирования воздушной среды в помещении.

Часть иерархических уровней может отсутствовать в зависимости от конкретного конструктивного исполнения ЭС, целей проектирования и протекающих в ЭС тепловых процессов.

Особенностью иерархии ЭС является, в свою очередь, иерархия математических моделей и задач, которые решаются при тепловом проектировании ЭС на каждом функционально-конструктивном уровне.

Тепловая модель ЭС на различных уровнях иерархии

Иерархическая структура ЭС (рис. 1) порождает иерархию взаимосвязанных задач, решаемых при тепловом проектировании ЭС. Их особенность в том, что решаются они последовательно, начиная с самого верхнего (пятого) уровня иерархии и заканчивая задачами низшего (первого) уровня, причем параметры, определяющие тепловые процессы в ЭС на более высоком иерархическом уровне, служат исходными входными данными (граничными условиями в математических моделях) для определения параметров теплового процесса на следующем, более низком, иерархическом уровне [9, 10].

Моделирование тепловых процессов ЭС требует построения тепловой модели на каждом иерархическом уровне ЭС. С этой целью сложную трехмерную конструкцию ЭС, состоящую из большого числа разнообразных и разнородных элементов, структурируют на иерархические уровни и переходят к рассмотрению конкретного уровня. На

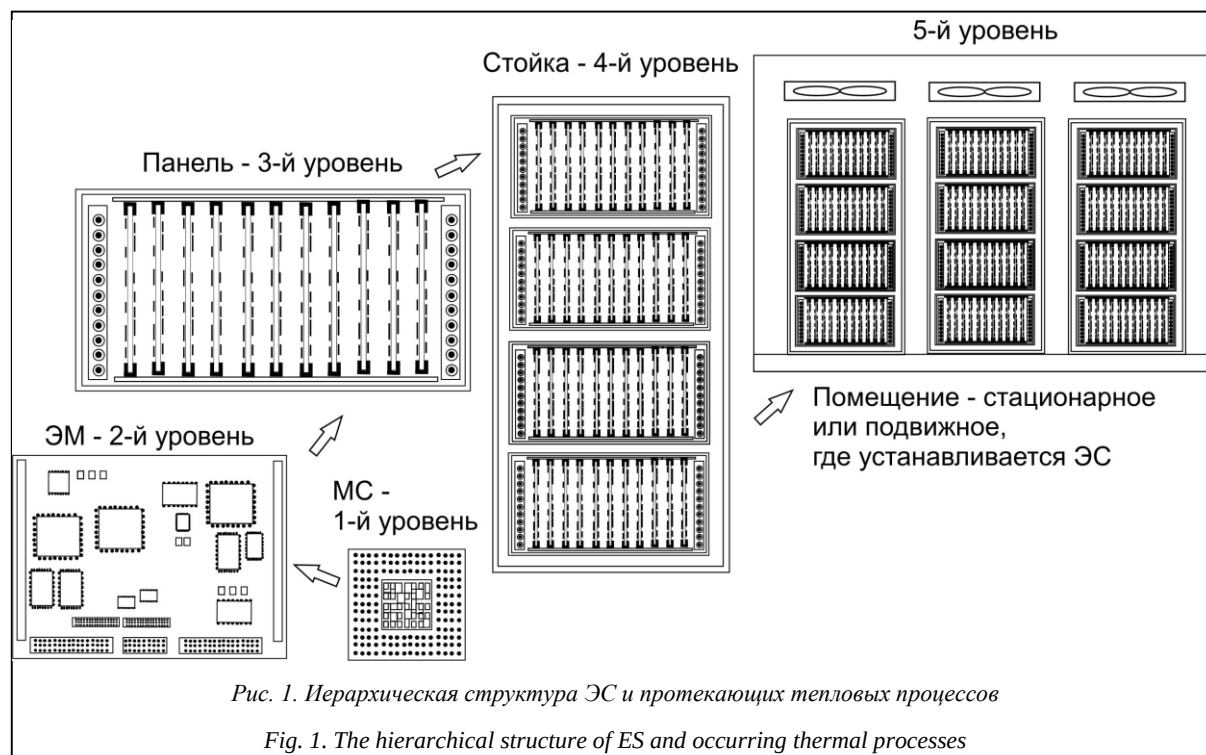


Рис. 1. Иерархическая структура ЭС и протекающих тепловых процессов

Fig. 1. The hierarchical structure of ES and occurring thermal processes

каждом уровне иерархии ЭС подвергается дискретизации на отдельные элементы, которая продолжается до тех пор, пока размеры и форма получаемых при этом элементов позволят с достаточной для практики точностью принять, что температурные поля внутри и на поверхности полученных дискретных элементов являются изотермическими.

Полученная таким образом система $\bigcup_{i=1}^N G_i$ из N изотермических и взаимодействующих между собой элементов G_i с границами ∂G_i образует тепловую модель ЭС (рис. 2) на данном иерархическом уровне.

Отметим, что детализация разбиения конструкции ЭС на элементы должна соответствовать конкретному рассматриваемому иерархическому уровню, а также требуемой точности расчета теплового процесса, сложности конструкции и целям дальнейшего использования полученных результатов для моделирования тепловых процессов остальных иерархических уровней.

Элементы в тепловой модели можно подразделить на три группы.

- **Активные элементы** – элементы, в которых происходит тепловыделение при потреблении энергии от внешних источников питания (на рисунке 2 элементы 1 и i с мощностью тепловыделения P_{s1} и P_{si}). К ним относятся такие элементы, как микросхемы, микропроцессоры, резисторы, катушки индуктивности, трансформаторы и другие электрорадиоэлементы, в которых происходит переход потребляемой энергии в тепловую. Активные элементы являются источником теплоты в тепловой модели.

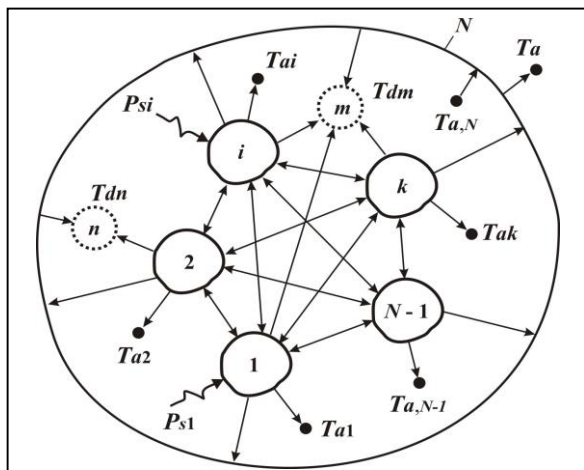


Рис. 2. Тепловая модель ЭС на иерархическом уровне в виде системы N изотермических тел, находящихся во взаимодействии друг с другом, средой внутри и вне ЭС и стоками теплоты

Fig. 2. ES thermal model on the hierarchical level in the form of N isothermal bodies, which interact with each other, the environment inside and outside of the ES and heat sinks

- **Пассивные элементы** – элементы (на рисунке 2 элементы 2, k , N), в которых происходит рассеивание теплоты (конструктивные элементы, электрорадиоэлементы без тепловыделения), участвующие в теплообмене с другими элементами ЭС (элемент N), в том числе и со стоками теплоты. В последнем случае элементы являются теплоотводами (элементы 2 и k).

- **Стоки теплоты** – элементы (на рисунке 2 элементы n и m с температурами T_{dn} и T_{dm} показаны пунктиром) с принудительно поддерживаемой температурой, которая значительно ниже температуры среды внутри ЭС, а также элементы, в которых осуществляется интенсификация теплообмена. К стокам относятся элементы системы охлаждения (каналы с хладоносителями, устройства с охлаждающими эффектами, вентиляторы, жидкостно-проточные и проточные системы и др.).

Отметим, что если температура окружающей среды как вне (на рисунке 1 жирная точка с температурой T_a), так и внутри ЭС возле каждого элемента (на рисунке 1 жирные точки с температурами T_{a1} , T_{a2} , ..., $T_{a,N}$) неизвестна и определяется априори неизвестными температурами элементов ЭС, то она в этом случае также включается в тепловую модель в виде самостоятельного изотермического элемента со своим номером. Если температура среды возле каждого элемента известна, замена ее элементом в тепловой модели не требуется.

Математическая модель теплового процесса в тепловой модели ЭС

Детерминированное распределение температуры в техническом объекте однозначно определяется следующими факторами (так называемыми условиями однозначности): формой и размерами объекта и его элементов, граничными условиями и входящими в них теплофизическими параметрами (коэффициентами теплоотдачи, температурой среды в объекте и вне его, плотностью тепловых потоков на границе), начальными условиями, распределением и интенсивностью внутренних источников теплоты, теплофизическими характеристиками материалов объекта и среды (плотностью, теплопроводностью, теплоемкостью, вязкостью). Когда определяющие факторы теплового процесса известны и однозначны, распределение температуры и описывающая его математическая модель являются детерминированными. Если хотя бы один из определяющих факторов теплового режима интервально стохастический, распределение температуры в техническом объекте тоже будет интервально стохастическим, а его математическая модель – стохастической.

Поскольку, как показано выше, тепловые процессы в ЭС являются нелинейными, динамическими (нестационарными), трехмерными и интервально стохастическими, то и математическая мо-

дель, описывающая тепловые процессы в ЭС, должна носить нелинейный, нестационарный, трехмерный, стохастический характер.

Математическая модель теплового процесса в ЭС, ее тепловой модели, основывается на интегральном балансовом уравнении [5, 12], описывающем теплообмен элементов системы между собой и средой в каждый момент времени t для каждого $\omega \in \Omega$, где (Ω, U, P) – вероятностное пространство: Ω – множество или пространство элементарных событий ω , U – σ -алгебра подмножеств Ω , P – вероятность на U . В результате стохастическая математическая модель теплового процесса ЭС для каждого элемента $i = 1, 2, \dots, N$ в тепловой модели ЭС запишется в виде $((t, \omega) \in [0, \tau] \times \Omega)$:

$$h_i \frac{dT_i(t, \omega)}{dt} + \sum_{j=1, i \neq j}^N J_{ij}(t, \omega) + J_{a,i}(t, \omega) = \Phi_i(t, \omega), \quad (1)$$

где $T_i(t, \omega)$ – стохастическая нестационарная температура i -го элемента, постоянная по объему V_i и ограничивающей поверхности S_i элемента; $J_{ij}(t, \omega)$ – стохастические тепловые потоки, поступающие от i -го элемента к остальным элементам, $j = 1, 2, \dots, i-1, i+1, \dots, N$; $J_{a,i}(t, \omega)$ – стохастический тепловой поток от элемента i в окружающую его среду; $\Phi_i(t, \omega)$ – интенсивность стохастического внутреннего источника теплоты в i -м элементе; $h_i = \rho_i c_i V_i$ – полная теплоемкость в объеме i -го элемента V_i ; ρ_i , c_i – плотность и теплоемкость материала i -го элемента.

Рассмотрим выражение для полного теплового потока $J_i(t, \omega)$ с поверхности элемента i ко всем элементам тепловой модели, в том числе и в среду:

$$J_i(t, \omega) = \sum_{j=1, i \neq j}^N J_{ij}(t, \omega) + J_{a,i}(t, \omega). \quad (2)$$

Поток теплоты J_{ij} между двумя изотермическими элементами i и j в общем случае может переноситься одновременно потоками кондукции J_{ij}^{cond} , конвекции J_{ij}^{conv} и излучения J_{ij}^{rad} , так что величина теплового потока $J_{ij} = J_{ij}^{cond} + J_{ij}^{conv} + J_{ij}^{rad}$. Перенос теплового потока с поверхности элемента i в среду $J_{a,i}$, в отличие от теплового потока J_{ij} , может осуществляться одновременно двумя потоками – потоком конвекции в среду возле элемента $J_{a,i}^{conv}$ и потоком излучения с поверхности элемента $J_{a,i}^{rad}$, то есть $J_{a,i} = J_{a,i}^{conv} + J_{a,i}^{rad}$.

Особенность электронных элементов ЭС в том, что разность температур между элементами в ЭС и между элементами и средой не превышает 150 °С, поэтому при моделировании тепловых процессов в ЭС с достаточной для практики точностью можно принять, что тепловые потоки $J_{ij}^{(cond, conv, rad)}$ и $J_{a,i}^{(conv, rad)}$ между элементом i и средой могут быть представлены в виде $J_{ij}^{(cond, conv, rad)} = g_{ij}^{(cond, conv, rad)}(T_i - T_j)$ и $J_{a,i}^{(conv, rad)} = g_{a,i}^{(conv, rad)}(T_i - T_{a,i})$, где $g_{ij}^{(cond, conv, rad)}$ – теп-

ловые проводимости (кондукции, конвекции, излучения) между элементами i и j с температурами T_i и T_j и между элементом i и средой с температурой $T_{a,i}$ [6, 11, 13–15].

Тепловые проводимости $g_{ij}^{(cond, conv, rad)}$ для процессов кондукции, конвекции (естественной и вынужденной) и излучения в ЭС подчиняются следующим закономерностям:

– кондуктивная тепловая проводимость $g_{ij} = g_{ij}^{cond}$ между элементами i и j определяется размерами, формой, теплопроводностью области распространения кондуктивного теплового потока, площадью сечения, пронизываемого тепловым потоком; зависимость кондуктивной проводимости от температуры практически отсутствует, за исключением, возможно, незначительной кондуктивной проводимости в газе и жидкости;

– конвективная тепловая проводимость $g_{ij}^{conv} = \alpha_{ij}^{conv} S_{ij}^{conv}$; коэффициент теплоотдачи α_{ij}^{conv} моделирует конвективный естественный и вынужденный теплообмен как между элементами, так и от элементов в среду; для естественной конвекции характерна следующая зависимость коэффициента теплоотдачи от разности температур между элементами или между элементами и средой (см. замечание выше): $\alpha_{ij}^{conv} = C(T_i - T_j)^n$, где $0 \leq n < 1$ – показатель степени, определяемый режимом протекания естественной конвекции (ламинарный, турбулентный, переходный); S_{ij}^{conv} – площадь элемента i , участвующего в конвективном теплообмене с элементом j ; C – коэффициент, зависящий от теплофизических характеристик среды, конструктивных факторов, ориентации теплоотдающей поверхности и пр.; для вынужденной конвекции при ламинарном переходном и турбулентном режимах коэффициент теплоотдачи α_{ij}^{conv} слабо зависит от температуры и определяется в основном площадью сечения потока, теплофизическими характеристиками потока среды (зависят от температуры), его скоростью;

– тепловая проводимость теплообмена излучением $g_{ij}^{rad} = \alpha_{ij}^{rad} \cdot S_{ij}^{rad}$; коэффициент теплоотдачи излучением α_{ij}^{rad} определяется выражением $\alpha_{ij}^{rad} = A(T_i^4 - T_j^4)/(T_i - T_j)$, где $A = \varepsilon_{ij} \Phi_{ij} \sigma$; ε_{ij} , Φ_{ij} – приведенная степень черноты и коэффициент облученности тел i и j соответственно; S_{ij}^{rad} – взаимная поверхность излучения элементов i и j ; $\sigma = 5,67 \cdot 10^{-8}$ Вт/м² К⁴ – постоянная Стефана–Больцмана.

Выражения для тепловых проводимостей конвекцией, излучением и коэффициентов теплоотдачи остаются в силе и для тепловых проводимо-

стей $g_{a,i}^{(conv,rad)}$, и для коэффициентов теплоотдачи $\alpha_{a,i}^{(conv,rad)}$ от элементов в среду, с той лишь разницей, что температуры T_j в последних заменяются на температуру среды $T_{a,i}$.

Потоки теплоты J_{ij} , J_{ij}^{cond} , J_{ij}^{conv} , J_{ij}^{rad} , $J_{a,i}$ нелинейно зависят от случайных температур элементов $T_i = T_i(t, \omega)$, $T_j = T_j(t, \omega)$ и среды $T_{a,i} = T_{a,i}(t, \omega)$, $i, j = 1, 2, \dots, N$, а также случайных величин кондуктивных проводимостей $g_{ij}(\omega) = g_{ij}^{cond}(\omega)$, поэтому тепловые потоки J_{ij} и $J_{a,i}$ в полном тепловом потоке J_i (2) являются нелинейными интервально стохастическими функциями от факторов $T_i(t, \omega)$, $T_j(t, \omega)$, $T_{a,i}(t, \omega)$, $g_{ij}(\omega)$, $\omega \in \Omega$.

Для учета влияния эффекта тепловой обратной связи в элементах ЭС на тепловой и электрический режимы ЭС необходимо ввести в функцию мощности потребления каждого элемента ее зависимость от температуры. Это приводит к тому, что функция мощности внутренних источников теплоты элементов Φ_i в (1) будет не только нестационарной и интервально стохастической, но и нелинейной, то есть $\Phi_i = \Phi_i(T_i, t, \omega)$.

Таким образом, математическая модель тепловых процессов в тепловой модели ЭС имеет вид $((t, \omega) \in [0, \tau] \times \Omega, i, j = 1, 2, \dots, N)$:

$$h_i \frac{dT_i(t, \omega)}{dt} + \sum_{j=1, j \neq i}^N J_{ij}(T_i, T_j, g_{ij}, t, \omega) + J_{a,i}(T_i, T_{a,i}, t, \omega) = \Phi_i(T_i, t, \omega) \quad (3)$$

со стохастическими начальными условиями

$$T_i(t=0, \omega) = T_{0,i}(\omega), \omega \in \Omega, \quad (4)$$

где $T_{0,i}(\omega)$ – начальные случайные температуры элементов $i = 1, 2, \dots, N$.

Уравнения (3) с начальными условиями (4) представляют собой систему нестационарных, нелинейных, интервально стохастических дифференциальных уравнений в обыкновенных производных первого порядка, полностью определяющих тепловые процессы в ЭС и описывающих нестационарные интервально стохастические температуры $T_i = T_i(t, \omega)$ элементов $i = 1, 2, \dots, N$ в тепловой модели ЭС.

Моделирование тепловых процессов в ЭС и их элементах

Математическая модель (3), (4) описывает нелинейные нестационарные стохастические тепловые процессы, развивающиеся на каждом отдельном уровне иерархии ЭС. Причем уравнения (3) математической модели учитывают как эффект тепловой обратной связи в элементах ЭС, так и интервально стохастическую неопределенность электрических и тепловых параметров элементов, обусловленную статистическим технологическим разбросом изготовления и монтажа элементов в ЭС.

Моделирование эффекта тепловой обратной связи (thermal feedback, self-heating effect) учитывается в уравнениях (3) наличием зависимости мощности потребления $\Phi_i(T_i, t, \omega)$ активного элемента тепловой модели ЭС от температуры элемента T_i . При этом последняя определяется как собственным разогревом, так и нагреванием в результате теплового взаимодействия со стороны остальных тел в системе. В силу изменения потребляемой мощности $\Phi_i(T_i, t, \omega)$ в зависимости от температуры, а также изменения температуры элемента T_i в результате изменений мощности $\Phi_i(T_i, t, \omega)$ происходит взаимообусловленное воздействие температуры на электрические параметры элемента ЭС, а электрических параметров – на температуру элементов ЭС. Такое интерактивное взаимодействие электрических и тепловых режимов характерно для электронных систем, элементная база которых выполняется на основе полупроводниковых материалов, обладающих повышенной чувствительностью к температурным изменениям и уровням температур. Зависимости мощностей потребления различных элементов ЭС от температуры приводятся в справочных данных на конкретный элемент или могут быть получены путем измерений. Необходимо отметить, что тепловая обратная связь может быть как положительной, так и отрицательной. В первом случае будет наблюдаться лавинообразный рост мощности потребления, вызывающий неограниченный рост температуры, что приведет к перегоранию элемента. Во втором случае, то есть при отрицательной тепловой обратной связи, со временем произойдет установление стационарного электрического и теплового режимов элементов и ЭС в целом.

Фактические данные по статистическому разбросу как электрических, так и тепловых параметров различных элементов ЭС, обуславливающих интервально стохастический характер тепловых и электрических режимов ЭС, описываемых стохастическими уравнениями (3), могут быть получены как у производителя элементов, так и в результате выборочного контроля партии элементов в лабораторных условиях [8].

Математическая модель (3), (4), как уже указывалось, описывает эволюцию тепловых процессов в ЭС на каждом иерархическом уровне ЭС (рис. 1). При этом результаты, полученные из расчета теплового процесса на данном иерархическом уровне, являются входными исходными данными для расчета теплового процесса следующего за данным более низкого иерархического уровня.

В практике теплового проектирования, помимо анализа теплового процесса на том или ином иерархическом уровне ЭС, возникает необходимость в более детальном расчете теплового режима собственно конкретного электронного элемента, который на данном уровне иерархии моделировался как изотермический. Так, например, получив

из теплового расчета панели (3-й уровень иерархии ЭС, рис. 1) температурное распределение воздушной среды и коэффициенты теплоотдачи внутри панели с поверхностей электронных модулей в среду, причем для каждого электронного модуля в панели, можно приступить к детальному моделированию теплового режима конкретного электронного модуля. Моделирование теплового режима электронного модуля осуществляется посредством подробного моделирования трехмерного температурного поля всей конструкции электронного модуля, которая включает в себя МПП, установленные на ней микропроцессоры, микросхемы в корпусах, учитывая при этом конструкции корпусов микросхем, электрорадиоэлементы, конструктивные элементы крепления и монтажа, электрические разъемы, установленные на МПП, теплоотводы, элементы системы охлаждения электронного модуля и др. Более детальное моделирование температурных полей элементов ЭС на том или ином иерархическом уровне может потребовать наряду с рассматриваемыми методами моделирования и иные методы математического и компьютерного моделирования (сеток, граничных/конечных элементов) [5, 8, 16, 17].

Поэтому при разработке МФПК-ТП-ЭС, предназначенного для теплового проектирования ЭС, необходимо включать в него как математические модели и алгоритмы развития тепловых процессов на различных иерархических уровнях, так и программные модули для детального моделирования температурных полей различных элементов ЭС. При создании МФПК-ТП-ЭС важнейшими задачами являются также разработки математических методов, позволяющих эффективно и гарантированно получать искомое решение системы уравнений математической модели (3), (4), и эффективных вычислительных алгоритмов, с наименьшими затратами машинного времени и памяти реализующими разработанные математические методы.

Собственно МФПК-ТП-ЭС должен содержать высокоэффективное вычислительное ядро и развитую сервисную пользовательскую оболочку, отвечающую современным требованиям к сложным программным комплексам и системам [18–20], обеспечивающую визуальную, наглядную и удобную для пользователя форму задания исходных данных, представление полученных результатов, визуализацию результатов в виде цветных изображений и изотерм полей температуры на различных иерархических уровнях ЭС.

В заключение отметим, что концепции теплового проектирования ЭС, изложенные в статье, создают реальную возможность для математического и компьютерного моделирования сложных тепловых процессов в ЭС и их элементах и осуществления всего цикла теплового проектирования ЭС, содержащих большое количество разнообразных и разнородных элементов. Полученная в статье мате-

матическая модель для моделирования тепловых процессов на различных иерархических уровнях ЭС учитывает различные аспекты тепловых процессов, с которыми сталкивается разработчик при создании и проектировании ЭС, а именно: нелинейный, динамический и нестационарный характер температурных распределений в элементах ЭС и в ЭС в целом; разброс электрических и тепловых параметров элементов ЭС; эффект тепловой обратной связи; стохастический характер температурных распределений ЭС. Разработанные в статье концепции и математические модели могут быть положены в основу МФПК-ТП-ЭС, предназначенного для теплового проектирования ЭС.

Литература

1. Hossain M. Thermal effects on analog integrated circuit design. The University of Texas at Arlington, 2007, 152 p.
2. Tesi di Laurea Modeling and Simulation of Self-Heating Effects in Deep Sub-Micron Silicon on Insulator (SOI) Technologies for Analog Circuits. Universita Degli Studi di Udine, 2004, 120 p.
3. Конструкторско-технологическое проектирование электронной аппаратуры; [под общ. ред. В.А. Шахнова]. М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. 568 с.
4. Чеканов А.Н. Расчеты и обеспечение надежности электронной аппаратуры. М.: КНОРУС, 2012. 440 с.
5. Мадера А.Г. Моделирование теплообмена в технических системах. М.: Изд-во НФ «Первая исслед. лаб. им. акад. В.А. Мельникова», 2005. 208 с.
6. Ellison G.N. Thermal computations for electronics. Conductive, radiative, and convective air cooling. NY, CRC Press, 2011, 416 p.
7. Бобков С.Г., Мадера А.Г. Энергетические затраты, быстродействие и проблема теплоотвода в микропроцессорах // Программные продукты и системы. 2013. № 4. С. 29–35.
8. Мадера А.Г., Кандалов П.И. Компьютерное моделирование температурных полей технических систем при интервально стохастической неопределенности параметров // Прикладная информатика. 2015. № 1 (55). С. 106–113.
9. Резников Г.В. Расчет и конструирование систем охлаждения ЭВМ. М.: Радио и связь, 1988. 224 с.
10. Мадера А.Г. Иерархический подход при тепловом проектировании электронных изделий // Программные продукты и системы. 2008. № 4 (84). С. 43–46.
11. Дульнев Г.Н., Тарновский Н.Н. Тепловые режимы электронной аппаратуры. Л.: Энергия, 1971. 248 с.
12. Дьярмати И. Неравновесная термодинамика. Теория поля и вариационные принципы. М.: Мир, 1974. 304 с.
13. Дульнев Г.Н. Тепло- и массообмен в радиоэлектронной аппаратуре. М.: Высш. шк., 1984. 247 с.
14. Михеев М.А., Михеева И.М. Основы теплопередачи. М.: БАСТЕТ, 2010. 344 с.
15. Heat exchanger design handbook. Heat exchanger theory. NY. Hemisphere Publ. Corp., 1983, 561 p.
16. Кандалов П.И., Мадера А.Г. Моделирование 3D-температурных полей электронных систем: программный комплекс STF-ElectronMod // Труды НИИСИ РАН. 2012. Т. 2. № 2. С. 67–69.
17. Kandalov P.I., Madera A.G. Mathematical and computing modeling of temperature fields in electronic modules. Proc. 16th Intern. Workshop on Thermal Investigations of ICs and Systems, IEEE. Barcelona, 2010.
18. Решетников В.Н. Космические телекоммуникации. Системы спутниковой связи и навигации. СПб: Изд-во РИПСРБ, 2010. 138 с.
19. Ли К. Основы САПР (CAD/CAM/CAE). СПб: Питер, 2004. 560 с.
20. Решетников В.Н., Мамросенко К.А. Основы построения тренажерно-обучающих систем сложных технических комплексов // Программные продукты и системы. 2011. № 3. С. 86–90.

DOI: 10.15827/0236-235X.112.079-086

Received 27.07.15

THE CONCEPT OF MATHEMATICAL AND COMPUTER SIMULATION OF THERMAL PROCESSES IN ELECTRONIC SYSTEMS

Madera A.G., Dr.Sc. (Engineering), Professor, Head of Department, alexmadera@mail.ru
(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation)

Abstract. The article presents the concepts of electronic systems thermal design, as well as the requirements to mathematical and computer simulation of thermal processes in electronic systems. The author formulates the requirements to the mathematical models, which provide adequate simulation of thermal processes when thermal designing of electronic systems. According to these requirements, the mathematical models must be dynamic, nonlinear, stochastic in intervals, consider the effect of thermal feedback (interaction between thermal and electrical modes in the electronic system) as well as statistical technology spread of electric and thermal defining parameters of elements in the electronic system. The obtained equations of the mathematical model for thermal processes in electronic systems meet the stated requirements. They are a system of the time-dependent, nonlinear, stochastic interval ordinary first-order derivatives. The developed concepts and mathematical models may be taken as the base of the multi-functional software package for thermal design of electronic systems.

Keywords: electronic system, thermal process, mathematical model, computer simulation, dynamic, nonlinear, interval, stochastic, thermal feedback.

References

1. Hossain M. *Thermal effects on analog integrated circuit design*. The Univ. of Texas at Arlington Publ., 2007, 152 p.
2. *Modeling and Simulation of Self-Heating Effects in Deep Sub-Micron Silicon on Insulator (SOI) Technologies for Analog Circuits*. Thesis. Univ. of Udine Publ., 2004, 120 p.
3. *Konstruktorско-tekhnologicheskoe proektirovanie elektronnoy apparatury* [Design and Engineering of Electronic Equipment]. V.A. Shakhnov (Ed.). Moscow, MSTU Publ., 2005.
4. Chekanov A.N. *Raschety i obespechenie nadezhnosti elektronnoy apparatury* [Calculations and Reliability Control of Electronic Equipment]. Moscow, KNORUS Publ., 2012.
5. Madera A.G. *Modelirovanie teploobmena v tekhnicheskikh sistemakh* [Heat Exchange Modeling in Engineering Systems]. Moscow, "Pervaya issled. Lab. im. akad. V.A. Melnikova" Publ., 2005.
6. Ellison G.N. *Thermal Computations for Electronics. Conductive, Radiative, and Convective Air Cooling*. NY, CRC Press, 2011.
7. Bobkov S.G., Madera A.G. Energy costs, speed and heat sink in microprocessors. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 4, pp. 29–35 (in Russ.).
8. Madera A.G., Kandalov P.I. Komputernoe modelirovanie temperaturnykh pol'y tekhnicheskikh sistem pri intervalno stokhasticheskoy neopredelennosti parametrov. *Prikladnaya informatika* [Applied Informatics]. 2015, no. 1 (55), pp. 106–113 (in Russ.).
9. Reznikov G.V. *Raschet i konstruirovaniye sistem okhlazhdeniya EVM* [Calculation and Design of Computer Cooling Systems]. Moscow, Radio i svyaz Publ., 1988.
10. Madera A.G. A Hierarchical Approach in heat design of electronic components. *Programmnye produkty i sistemy* [Software & Systems]. 2008, no. 4 (84), pp. 43–46.
11. Dulnev G.N. *Teplovye rezhimy elektronnoy apparatury* [Heating Conditions of Electronic Equipment]. Leningrad, Energiya Publ., 1971.
12. Dyarmati I. *Neravnovesnaya termodinamika. Teoriya polya i variatsionnye printsipy* [Non-equilibrium Thermodynamics. Field Theory and Variation Principles]. Moscow, "Mir" Publ., 1974.
13. Dulnev G.N. *Teplo- i massoobmen v radioelektronnoy apparature* [Heat and Mass Exchange in Electronic Facilities]. Moscow, Vyssh. Shk. Publ., 1984.
14. Mikheev M.A., Mikheeva I.M. *Osnovy teploperedachi* [Basics of Heat Transfer]. Moscow, BASTET Publ., 2010.
15. *Heat exchanger design handbook. Heat exchanger theory*. NY, Hemisphere Publ. Corp., 1983.
16. Kandalov P.I., Madera A.G. Modeling of 3D temperature fields of electronic systems: STF-ElectronMod software package. *Trudy NIISI RAN* [Proc. of SRISA RAS]. 2012, vol. 2, no. 2, pp. 67–69 (in Russ.).
17. Kandalov P.I., Madera A.G. Mathematical and computing modeling of temperature fields in electronic modules. *Proc. of the 16th Int. Workshop on Thermal Investigations of ICs and Systems, THRMINIC 2010*. IEEE. Barcelona, 2010.
18. Reshetnikov V.N. *Kosmicheskije telekommunikatsii. Sistemy sputnikovoy svyazi i navigatsii* [Space Telecommunications. Satellite Communication Systems and Navigation Systems]. St. Petersburg, RIPDPb Publ., 2010, 138 p.
19. Lee K. *Principles of CAD/CAM/CAE systems*. NY, Addison-Wesley Publ., 2004, 560 p..
20. Reshetnikov V.N., Mamrosenko K.A. The framework of simulation training systems of complex slozhnykh technical sets. *Programmnye produkty i sistemy* [Software & Systems]. 2011, no. 3, pp. 86–90.

УДК 004.716

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.087-092

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ БЛОК ИНТЕРФЕЙСА RapidIO ДЛЯ СОЗДАНИЯ МНОГОЯДЕРНЫХ МИКРОПРОЦЕССОРОВ С ВИРТУАЛЬНЫМИ КАНАЛАМИ RapidIO

*Н.А. Козлов, аспирант, младший научный сотрудник, kozlov_n@cs.niisi.ras.ru
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия);*

*С.Г. Бобков, д.т.н., профессор, зам. директора, bobkov@cs.niisi.ras.ru
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия;
Национальный исследовательский ядерный университет «МИФИ»,
Каширское шоссе, 31, г. Москва, 115409, Россия)*

Одним из основных направлений повышения производительности вычислительных систем является создание многоядерных и многопроцессорных систем. Современные многопроцессорные системы, как правило, строятся на базе коммуникационных сред. К наиболее распространенным высокопроизводительным средам относятся сети HyperTransport, PCIExpress, ASI, RapidIO, VXS, StarFabric, Ethernet 10 Gb, InfiniBand, Myrinet. Коммуникационные среды создаются на внутрикристалльном, модульном, межмодульном и межмашинном уровнях. Стандарт RapidIO ориентирован на создание трех последних сред и, таким образом, позволяет значительно унифицировать многопроцессорные системы. Для дальнейшей унификации с целью повышения надежности и снижения трудозатрат на разработку необходима унификация модуля обмена между ядром микропроцессора и внешней средой. В статье рассматривается унифицированный блок (блок RIO-AXI) перехода с внутривычислительной шины AXI на внешнюю шину RapidIO, позволяющий создавать как многоядерные процессоры, так и коммутаторы с коммуникационной средой RapidIO.

Ключевые слова: коммутатор, RapidIO, многоядерный процессор, виртуальный канал.

Основным показателем производительности вычислительной системы является производительность микропроцессора. Д. Паттерсон и Дж. Хеннесси [1] определили производительность микропроцессора как

$$\frac{N_{\text{инст}}}{\text{Программа}} \times \frac{N_{\text{цикл}}}{N_{\text{инст}}} \times \frac{T_{\text{прог}}}{N_{\text{цикл}}} = \frac{T_{\text{прог}}}{\text{Программа}} =$$

= время работы CPU.

Как видим, производительность зависит от трех характеристик: частоты, частоты на инструкцию и количества инструкций. Более того, время работы процессора (CPU) в равной степени зависит от каждой из них: увеличение на 10 % одной дает общий прирост тоже в 10 %. К сожалению, сложно изменить один из этих параметров отдельно от остальных, поскольку основные технологии, определяющие каждую характеристику, взаимозависимы: частота определяется технологией изготовления микропроцессора, частота на инструкцию – архитектурой микропроцессора, количество инструкций – системой команд и технологией компиляции.

Улучшение технологических норм изготовления кристаллов микросхем и увеличение числа транзисторов в микросхемах примерно до 2003 г. тоже вызывало постоянный рост частоты микропроцессоров вплоть до 3–4 ГГц. Однако дальнейшее повышение частоты обусловило существенное повышение потребления питания и, соответственно, повышение тепловыделения, которое невозможно отвести от кристаллов стандартными средствами [2].

Повышение производительности стало достигаться за счет усложнения архитектуры ядра микропроцессора, включения дополнительных функ-

ций, сопроцессоров и, главное, за счет создания многоядерных микропроцессоров и многопроцессорных систем [2]. В связи с этим на первый план стала выдвигаться задача создания высокоэффективной коммуникационной среды. Одной из наиболее перспективных сред является коммуникационная среда RapidIO (RIO). Стандарт RIO разрабатывался специально для удовлетворения важнейших требований приложений реального времени: обеспечения малых задержек, детерминизма, надежности и масштабируемости, снижения энергопотребления, размеров и веса, обусловленного требованиями встроенных систем. Высокие показатели по скорости передачи и надежности привели к тому, что RIO начинает использоваться и для построения серверов, прежде всего серверов с плотной упаковкой. Кроме того, стандарт RIO позволяет унифицировать коммуникационную среду на модульном, межмодульном и межмашинном уровнях [3], что существенно удешевляет систему и упрощает ПО. Для дальнейшей унификации и снижения затрат на разработку предложен блок (блок RIO-AXI) перехода с внутривычислительной шины AXI на внешнюю шину RIO, позволяющий создавать как многоядерные процессоры, так и коммутаторы с коммуникационной средой RIO.

Унифицированный блок межпроцессорного обмена

На рисунке 1 приведена структурная схема организации обмена процессорного ядра (ЦПУ) с внешней коммуникационной средой через каналы RIO. К процессорному ядру подключено *оперативное запоминающее устройство* (ОЗУ).

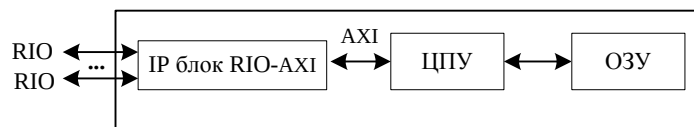


Рис. 1. Структурная схема взаимодействия процессорного ядра с коммуникационной средой RIO

Fig. 1. A flow chart showing interaction between processor core and RIO

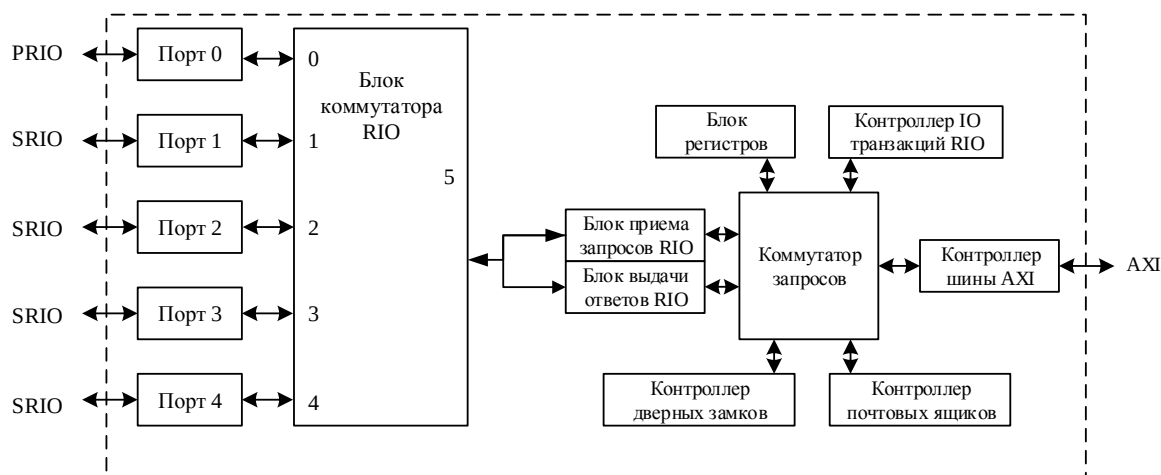


Рис. 2. Структурная схема блока межпроцессорного обмена ядра микропроцессора и среды RIO

Fig. 2. A flow chart for the block of interprocessor exchange of microprocessor core and RIO

На рисунке 2 приведена структурная схема блока межпроцессорного обмена. Схема состоит из следующих основных блоков: порты RIO, блок коммутации RIO и блок преобразования пакета RIO в пакет AXI. Блок преобразования состоит из блоков приема запроса и выдачи ответа RIO, коммутатора запросов, блока регистров, контроллера шины AXI и еще трех контроллеров: контроллера дверных замков, почтовых ящиков (контроллер сообщений) и транзакций ввода-вывода RIO.

Во время работы внешнее обращение поступает в порт RIO, далее пакет в зависимости от настроек таблицы маршрутизации передается в один из выходных портов. При обращении к процессору (попадании в порт 5) пакет RIO преобразуется в пакет AXI и затем поступает в процессор. С точки зрения программной модели, внутри микросхемы находится одно RIO-устройство.

Решение задачи построения блока межпроцессорного обмена для двух ядер

Одной из основных задач унификации среды является связь между двумя ядрами процессора на одном кристалле с сохранением доступа к периферийным устройствам (рис. 3). Рассмотрим характеристики, свойства и трудозатраты на реализацию блока межпроцессорного обмена для двух процессоров. Наиболее очевидными являются два варианта исполнения этого блока: подключение к одному порту RIO с использованием коммутатора AXI; подключение к порту RIO с созданием виртуальных каналов.

Первый вариант предполагает структуру, изображенную на рисунке 4. Рассмотрим затраты на разработку аппаратной части. Для реализации этой схемы необходимо разработать коммутатор AXI.

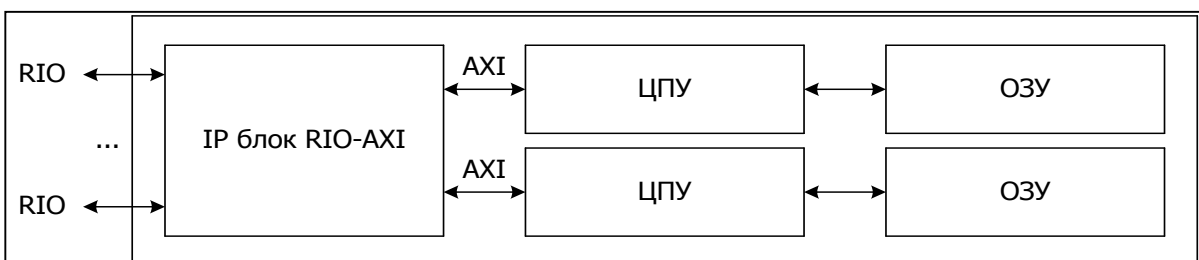


Рис. 3. Структурная схема взаимодействия двух процессоров с коммуникационной средой RIO

Fig. 3. A flow chart showing interaction between two processors and RIO

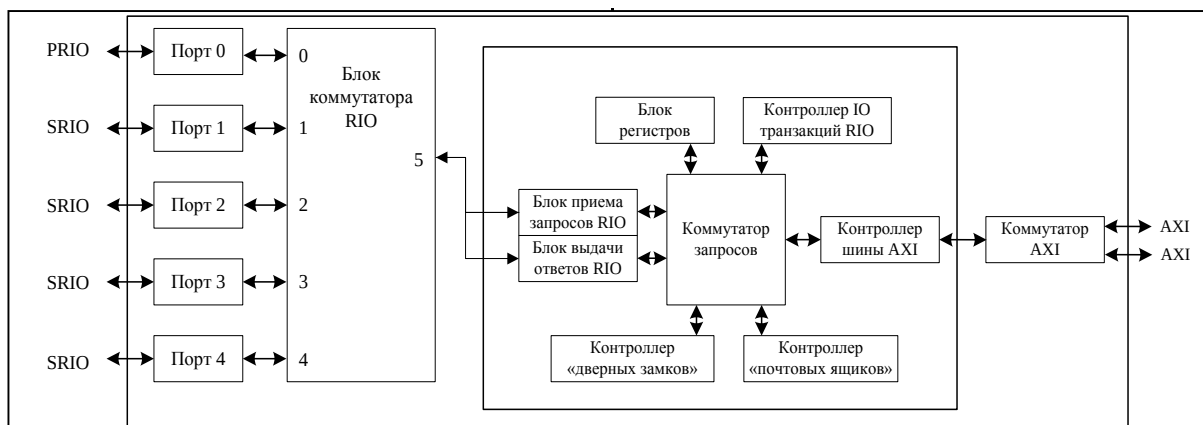


Рис. 4. Структурная схема блока межпроцессорного обмена для двух процессоров, вариант 1

Fig. 4. A flow chart showing interprocessor exchange for two processors, option 1

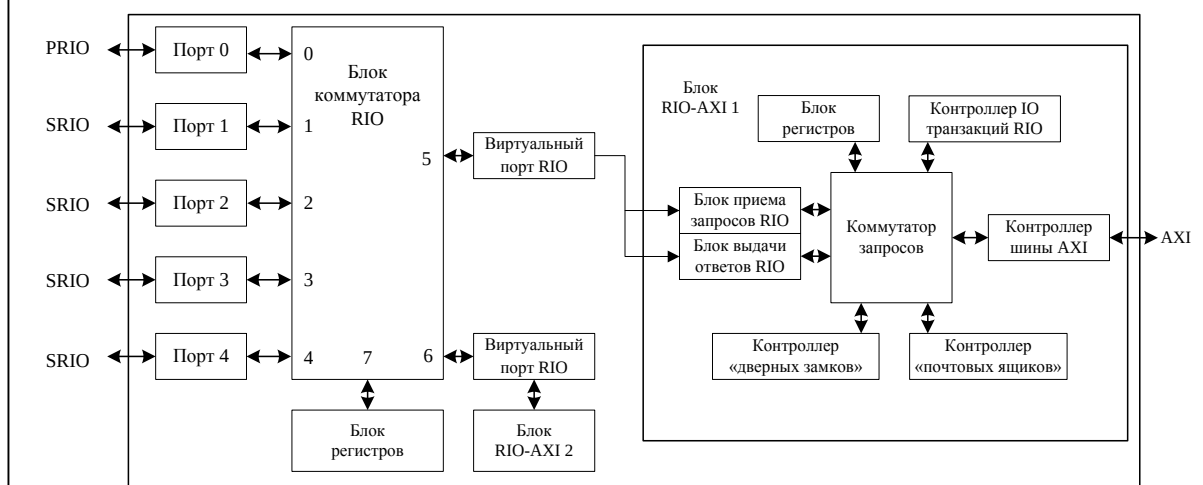


Рис. 5. Структурная схема блока межпроцессорного обмена для двух процессоров, вариант 2

Fig. 5. A flow chart showing interprocessor exchange for two processors, option 2

Далее при работе с одним портом RIO нужно разделить пакеты между двумя процессорами. Это можно реализовать через разделение адресного пространства, через контроллер сообщений или тип пакета. Для такого разделения необходимо значительно изменить контроллер сообщений. На изменение программной части требуется значительное изменение работы с разделением адресного пространства с двумя процессорами, находящимися на одном порту RIO. К достоинствам такого подхода можно отнести отсутствие дублирования контроллеров, к недостаткам – появление дополнительной задержки при прохождении пакета через AXI-коммутатор по сравнению со схемой, по которой происходит взаимодействие с одним ядром.

Для реализации второго способа соединения процессоров (рис. 5) необходимо сделать следующие изменения аппаратной части: подключить блок регистров к блоку коммутации, незначительно изменить структуру блока регистров AXI, подключить второй блок преобразования пакета

RIO в пакет AXI. В программной модели обозначенная структура будет определена как три устройства RIO, с каждым из которых можно работать независимо от другого: коммутатор RIO, блок RIO-AXI 1, блок RIO-AXI 2. При таком соединении соблюдается преемственность кода, все имеющиеся наработки с внесением незначительных изменений можно применить к новому блоку. Недостаток состоит в дублировании внутреннего содержания блока преобразования пакета RIO в AXI, достоинство – в сохранении задержки передачи на том же уровне, что и в унифицированном блоке с одним процессором.

Как видим, реализация второго варианта не только менее трудозатратна, но и обладает меньшими задержками при передаче пакета, являющимися одним из основных параметров таких систем. Таким образом, при относительно небольшом числе ядер данное решение оптимально. Принятие разработчиком решения должно основываться на оптимизации трудоемкости проекта, производи-

тельности, потребляемой мощности и числе транзисторов (площади) для данных вариантов. С учетом этих параметров выбран второй вариант.

Рассмотрим выбранную схему соединения процессоров. Схема состоит из трех блоков (рис. 5). Первый блок – это 8-портовый коммутатор, к которому подсоединен 1 порт PRIO, 4 порта SRIO, блок регистров и 2 виртуальных порта RIO. К первому виртуальному порту подключен блок RIO-AXI 1, который состоит из коммутатора запросов AXI, блока приема запросов и выдачи ответов RIO, блока регистров и трех контроллеров: дверных замков, почтовых ящиков и транзакций ввода-вывода RIO. Блок RIO-AXI 2 полностью аналогичен блоку RIO-AXI 1.

Опишем процесс передачи сообщения между процессорами, находящимися на одном кристалле. Примем, что таблица маршрутизации сконфигурирована. В начале передачи первый процессор, подключенный по AXI к блоку RIO-AXI 1, заполняет поля в блоке регистров. Во время этой операции в блок регистров AXI пишутся адрес, данные, тип пакета, возможное число переходов и другие служебные поля. После этого пакет поступает в блок преобразования из интерфейса AXI в интерфейс локальной шины. Затем пакет поступает в блок преобразования из локальной шины в интерфейс RIO – блок выдачи ответов RIO. После этой передачи пакет поступает на виртуальный канал RIO, а затем в 5-й порт коммутатора и передается в 6-й в соответствии с таблицей маршрутизации. После появления пакета на 6-м порту коммутатора описанная операция выполняется в обратном порядке, за исключением передачи данных из виртуального канала в блок приема ответов RIO и выхода пакета из блока RIO-AXI 2 на интерфейс AXI ко второму процессору. В случае передачи на соседний кристалл операция аналогична описанной, за исключением конфигурации таблицы маршрутизации.

Структура ядра коммутатора

Рассмотрим ядро коммутатора RIO. Коммутатор относится к классу коммутаторов со входной буферизацией [4] (Input-Queued Switch). Он может работать в двух режимах: приоритетной передачи пакетов и циклическом [5] (Round-robin). Если установлен циклический режим, то при передаче из нескольких входных портов N в один выходной пакеты передаются по очереди от порта $n = 1$ до N . Если коммутатор работает в режиме приоритетной передачи и несколько входных портов передают пакет в один выходной, то первым передается наиболее приоритетный пакет. Пакеты с равным приоритетом передаются в циклическом режиме.

Поскольку рассматриваемый коммутатор относится к классу коммутаторов со входной буферизацией, существует проблема блокировки головного пакета (HoL, Head-of-Line) [6]. Она решена путем

перемещения блокирующего пакета на случайное место в очереди. При этом перемещении выполняется также перемещение более приоритетных пакетов в начало очереди.

Данная архитектура коммутатора является оптимальной благодаря простоте реализации и хорошим временным характеристикам, а именно, низкой задержке прохождения пакета и, как следствие, низким аппаратным затратам. Недостатком такого решения является отсутствие алгоритма, находящего максимальные паросочетания (MWM, MaximumWeightMatching) [7], из-за чего выходные порты ядра коммутатора могут простаивать, понижая максимальную пропускную способность в случае неравномерного распределения номера порта назначения и длины пакетов.

Блок межпроцессорного обмена с интерфейсами RIO и AXI

Как было показано ранее, данную схему IP блока RIO-AXI реализовать достаточно просто при наличии всех его составных частей.

Рассмотрим достоинства созданного IP-блока.

- Возможность использования стандартного интерфейса передачи сообщений (Message Passing Interface, MPI), позволяющего процессорам обмениваться сообщениями, параллельно выполняя одну задачу [8]. Основной механизм – передача сообщений между процессорами.

- Низкая задержка передачи пакета. Задержка при передаче пакета между портами RIO, состоящая из задержки приемопередатчика (преобразование последовательного кода в параллельный 26 нс), преобразования в порту RIO (получение RIO-пакета из параллельного кода, одно преобразование 156 нс), буферизации во входных очередях (21 нс) и коммутации (14 нс) между портами, составляет 400 нс для технологии проектирования 250 нм.

- Масштабируемость. Возможно соединение 5 кристаллов без дополнительных элементов (см. рис. 6). Chip означает отдельный кристалл, CPU –

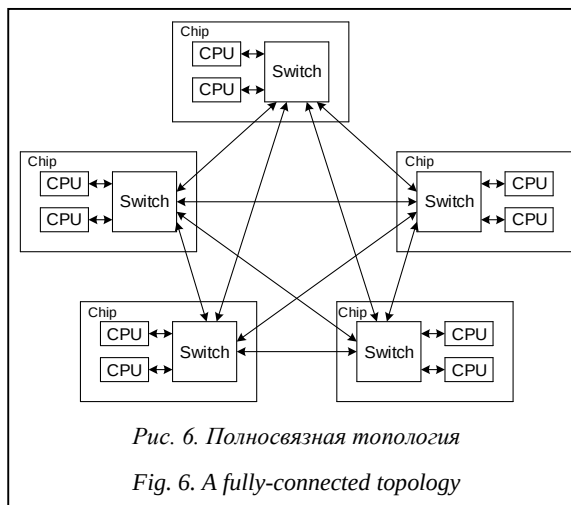
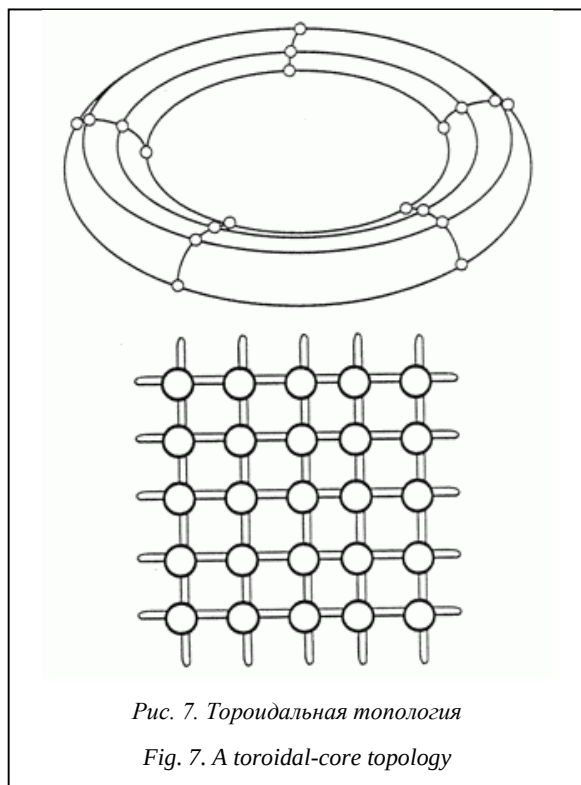


Рис. 6. Полносвязная топология

Fig. 6. A fully-connected topology



ядро одного процессора, Switch-IP – ядро RIO-AXI. При помощи дополнительных коммутаторов возможно соединение до 256 процессоров. Возможно соединение и более сложных многоядерных схем, например тороидальной сети (рис. 7).

- Совместимость с имеющимся ПО. Имеющиеся наработки ПО совместимы с этим IP-ядром, сохраняется модель контроллера передачи сообщений (message controller).

- Однородность доступа к процессору на одном и нескольких кристаллах. Для программиста обращение к процессору на одном кристалле выглядит так же, как и обращение к другому кристаллу.

- Дублирующий режим. В некоторых надежных системах существует требование выполнения операций процессорами в дублирующем режиме. Использование разработанного ядра позволяет поддерживать этот режим.

- Возможность дальнейшего развития. При необходимости блок можно расширить, увеличив количество портов коммутатора. В результате увеличивается количество периферийных портов или виртуальных каналов RIO для подключения процессоров. Программные наработки при этом также сохраняются.

После сборки и отладки проекта были проведены исследования временных параметров – измерение задержки при передаче пакетов от одного процессора к другому. Измерения задержки были выполнены на модели и составили порядка 400 нс. Полученные значения говорят о низкой задержке передачи пакета. С учетом поправки на рабочую тактовую частоту эта задержка сопоставима с задержкой, полученной авторами при тестировании многопроцессорного взаимодействия [9]. Таким образом, в результате проделанной работы определены возможные варианты создания унифицированных блоков каналов RIO, проведена оптимизация и разработан блок, обеспечивающий высокую производительность различных многопроцессорных систем.

Литература

1. Patterson D.A., Hennessy J.L. Computer architecture: a quantitative approach. 4th ed. Morgan Kaufmann Publ. Elsevier Science (ref.), 2011, 676 p.
2. Бобков С.Г. Высокопроизводительные вычислительные системы. М.: Изд-во НИИСИ РАН, 2014. 299 с.
3. Бобков С.Г., Задябин С.О. Перспективные высокопроизводительные вычислительные системы промышленного применения на базе стандарта RapidIO // Электроника, микро- и наноэлектроника: сб. науч. тр. 11-й Рос. науч.-технич. конф. (2009, г. Н. Новгород); [под ред. В.Я. Стенина]. М.: Изд-во МИФИ, 2009. С. 114–121.
4. McKeown N., Mekkittiku A., Anantharam V., Walrand J. Transactions on Communications, Aug. 1999, vol. 47, no. 8, pp. 1260–1267.
5. Silberschatz A., Galvin P.B., Gagne G. Process Scheduling. Operating System Concepts (8th ed.). John Wiley & Sons Publ. (Asia), 2010, 194 p.
6. Bennett J.C.R., Partridge C., Shtelman N. Packet reordering is not pathological network behavior. Dec. 1999, pp. 789–798.
7. Ерзин А.И. Введение в исследование операций: учеб. пособие. Новосибирск: Изд-во НГУ, 2006. С. 54–66.
8. Оленев Н.Н. Основы параллельного программирования в системе MPI. М.: Изд-во ВЦ РАН, 2005. 5 с.
9. Devashish P. Supercomputing Clusters with RapidIO Interconnect Fabric Ethernet Summit. 2015, April 14–16, 2015, p. 4.

DOI: 10.15827/0236-235X.112.087-092

Received 27.07.15

HIGH PERFORMANCE RAPIDIO BLOCK TO CREATE MULTI-CORE MICROPROCESSORS WITH RAPIDIO VIRTUAL LINKS

Kozlov N.A., Postgraduate Student, Junior Researcher, kozlov_n@cs.niisi.ras.ru

(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation);

Bobkov S.G., Dr.Sc. (Engineering), Professor, Deputy Director, bobkov@cs.niisi.ras.ru

(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation);

National Research Nuclear University «MEPhI», Kashirskoe Highway 31, Moscow, 115409, Russian Federation)

Abstract. One of the main methods of improving computing systems performance is the development of multi-core and multiprocessor systems. Modern multiprocessor systems are usually based on the communication environment. The most popular high-performance network environments are: HyperTransport, PCI Express, ASI, RapidIO, VXS,

StarFabric, 10GB Ethernet, InfiniBand, Myrinet. A communication environment is created on chip level, module level, inter-module level and inter-machine level. The RapidIO standard is focused on creating the last three environments and thus we can unify a multiprocessor system. For further unification we need to develop the module of the exchange between the microprocessor's core and the external environment in order to improve the reliability and reduce labor effort. The article discusses a unified adapter block (RIO-AXI) for a transition from on-chip AXI bus to the external RapidIO bus to create as multicore processors as well as switches with RapidIO environment.

Chip production technology improvement and increase in the number of transistors in the chips until 2003 has led to permanent growth of microprocessor frequency up to 3–4 GHz. However, further increase in frequency has caused a substantial increase of power consumption and heat dissipation, which is impossible to withdraw from crystals using standard means.

Increased productivity has been achieved by expenses on complicated architecture of microprocessor cores, additional functions, coprocessors and, most importantly, by creating multi-core microprocessors and multiprocessor systems. Therefore, a high-performance communication environment is emphasized. RapidIO communication environment is one of the most promising. The RapidIO standard is designed specifically to meet critical requirements of real-time applications, which are: low latency, determinism, reliability and scalability, reduction of power consumption, size and weight.

Keywords: switch, RapidIO, multi-core processor, virtual channel.

References

1. Patterson D.A., Hennessy J.L. *Computer Architecture: A Quantitative Approach*. 4th ed., Morgan Kaufmann Publ., Elsevier Science, 2011, 676 p.
2. Bobkov S.G. *Vysokoproizvoditelnye vychislitelnye systemy* [High-Performance Computer Systems]. Moscow, NIISI RAN Publ., 2014, 299 p.
3. Bobkov S.G., Zadyabin S.O. Promising industrial high-performance computer systems based on RapidIO standard. *Sbornik nauchnykh trudov 11 Ross. nauch.-tekhnich. konf. Elektronika, micro i nanoelectronika* [Proc. of the 11th Russian Science and Tech. Conf. Electronics, micro- i nanoelectronics]. Moscow, MIFI Publ., 2009, pp. 114–121 (in Russ.).
4. McKeown N., Mekkittiku A., Anantharam V., Walrand J. *Transactions on Communications*. 1999, vol. 47, no. 8, pp. 1260–1267.
5. Silberschatz A., Galvin P.B., Gagne G. *Process Scheduling. Operating System Concepts* (8th ed.). John Wiley & Sons (Asia) Publ., 2010, 194 p.
6. Bennett J.C.R., Partridge C., Shectman N. *Packet reordering is not pathological network behavior*. 1999, pp. 789–798.
7. Erzin A.I. *Vvedenie v issledovanie operatsy* [Introduction to Operation Analysis]. Study guide, Novosibirsk, Novosibirsk State Univ. Publ., 2006, pp. 54–66 (in Russ.).
8. Olenov N.N. *Osnovy paralelnogo programmirovaniya v sisteme MPI* [The Basics of parallel programming in MPI]. VTs RAN Publ., 2005, 5 p. (in Russ.).
9. Devashish P. *Supercomputing Clusters with RapidIO Interconnect Fabric Ethernet Summit*. 2015, April 14–16, 2015, p. 4.

Примеры оформления статьи в списке литературы

1. Козлов Н.А., Бобков С.Г. Высокопроизводительный блок интерфейса RapidIO для создания многоядерных микропроцессоров с виртуальными каналами RapidIO // Программные продукты и системы. 2015. № 4 (112). С. 88–92.
2. Козлов Н.А., Бобков С.Г. Высокопроизводительный блок интерфейса RapidIO для создания многоядерных микропроцессоров с виртуальными каналами RapidIO // Программные продукты и системы. 2015; DOI: 10.15827/0236-235X.112.088-092.
3. Kozlov N.A., Bobkov S.G. High performance RapidIO block to create multi-core microprocessors with RapidIO virtual links. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 4 (112), pp. 88–92 (in Russ.); DOI: 10.15827/0236-235X.112.088-092.

УДК 004.454

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.093-098

ПОДДЕРЖКА ПРОТОКОЛА MPI В ЯДРЕ ОС LINUX ДЛЯ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ НА БАЗЕ ВЫСОКОСКОРОСТНЫХ КАНАЛОВ RapidIO

*А.С. Кулешов, аспирант, rmdfax@cs.niisi.ras.ru
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

Решение вычислительных задач на многопроцессорных вычислительных комплексах в конечном итоге опирается на определенный программный интерфейс передачи данных. Таким наиболее распространенным интерфейсом является интерфейс передачи сообщений MPI, который определяет стандарт передачи данных для конечных пользовательских программ. В статье показана реализация протокола MPI для разработанной в НИИСИ РАН микросхемы 1890BM6Я, являющейся основой для многопроцессорных вычислительных комплексов различного назначения, узлы которого могут быть связаны через высокоскоростные каналы RapidIO. Для реализации были взяты наиболее распространенная библиотека MPI – MPICH и хорошо зарекомендовавшее себя в суперкомпьютерных вычислениях ядро ОС Linux. MPICH в базовой поставке предоставляет различные сетевые модули, реализующие связку интерфейса библиотеки с определенными транспортными драйверами: tcp (Ethernet TCP/IP), ib (Infiniband), mx (Myrinet eXpress) и другими. Задействование наиболее подходящего под архитектуру RapidIO сетевого модуля mx позволило сосредоточиться только на разработке Linux-драйвера для контроллера RapidIO. Особенности контроллера RapidIO микросхемы 1890BM6Я, интерфейса MPI и идея использования канала RapidIO для других, отличных от MPI целей позволили создать достаточно универсальный стек передачи сообщений через канал RapidIO без избыточных копирований данных. В конце статьи приведены результаты работы тестовых MPI-программ, таких как NAS Parallel Benchmarks и OSU Micro-Benchmarks, на 4 узлах через канал RapidIO, дано заключение о проделанной работе и подведен итог использования контроллера RapidIO микросхемы 1890BM6Я в качестве транспортной среды для протокола MPI.

Ключевые слова: RapidIO, MPI, MPICH, Linux.

Проблема построения эффективных вычислительных систем, решающих разнообразные задачи – от научных до военных, упирается в доставку данных до вычислительных компонент: будь то центральный процессор, специализированная интегральная схема или графический ускоритель. В то время как на аппаратном уровне идет бурное развитие различных сетевых идеологий, с программной точки зрения самым распространенным средством передачи данных для многопроцессорных комплексов является интерфейс передачи сообщений (MPI) [1].

Разработанная в НИИСИ РАН микросхема 1890BM6Я, являющаяся микросхемой общего назначения, имеет в своем составе, помимо всего прочего, центральный процессор, мощный математический сопроцессор и контроллер RapidIO [2]. Коммутатор RapidIO 1890КПЗ, также разработанный в НИИСИ РАН, позволяет объединить эти микросхемы в сеть RapidIO и сформировать высокопроизводительный многопроцессорный вычислительный комплекс. Далее в данной статье представлена реализация MPI для вычислительного комплекса на базе микросхем семейства 1890BM6Я с возможностью исполнения операций приема/передачи данных по RapidIO без копирования (zero-copy/data offload).

При решении задачи запуска MPI-приложений на вычислительном комплексе возникает один из главных вопросов – рабочее окружение, состоящее из таких компонентов, как ядро ОС и набор библиотек. В качестве ядра ОС было выбрано ядро Linux, так как оно в связке с библиотеками MPI по-

казало свою эффективность в распределенных вычислениях [3–5], а в качестве библиотеки MPI – MPICH как одна из широко используемых реализаций MPI [6].

Архитектура библиотеки MPICH

В процессе своего развития библиотека MPICH пришла к стройной многоуровневой архитектуре [7, 8] (см. рис. 1).

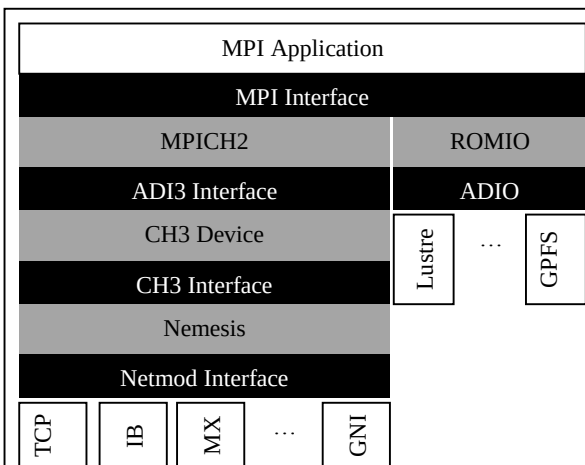


Рис. 1. Структура библиотеки MPICH:
темным фоном обозначен интерфейс,
светлым – код логического устройства

Fig. 1. MPICH library structure:
interface is in the dark background,
a logical device code is in the light background

В верхней части библиотеки находится совместимый со стандартом MPI версии 2.2 интерфейс, предоставляемый прикладным программам. Ниже по уровню располагается логическое устройство, реализующее абстрактный интерфейс устройства (Abstract Device Interface version 3 – ADI3 Interface). Портирование MPICH для определенной коммуникационной среды может заключаться в реализации такого ADI3-устройства (например, на рисунке 1 это CH3 Device). Данный подход позволяет наилучшим образом наложить интерфейс библиотеки на конкретное оборудование, однако процесс разработки и сопровождения полного ADI3-устройства может быть весьма трудоемким из-за своей сложности и постоянного развития верхнего уровня (поддержка более новых версий MPI может повлечь за собой введение новых или изменение старых функций).

Другой подход заключается в разработке устройства более низкого уровня, взаимодействующего с имеющимся в библиотеке MPICH ADI3 устройством (CH3 Device) по интерфейсу CH3I. Устройства такого уровня называются каналами. Естественным образом добавление нового уровня абстракции немного снижает производительность, но разработка и сопровождение такого канала требуют значительно меньше трудовых затрат. Однако практика показала, что разработчики вместо реализации каналов для конкретного оборудования перерабатывали само CH3-устройство. Стало ясно, что интерфейс CH3I не удовлетворял потребностям современных на то время технологий. Отсюда возникла необходимость в создании еще одного слоя абстракции, позволяющего стабилизировать CH3I-интерфейс.

Таким образом, был представлен канал Nemesis для CH3-устройства [9]. Ключевыми особенностями Nemesis являются масштабируемость, высокая производительность для внутриузловых и межузловых обменов, а также поддержка разнородных сетевых каналов передачи. Главные компоненты Nemesis – высокооптимизированная система сообщений для внутриузловых обменов и средства для реализации сетевых модулей (Network Modules – netmod), которые позволили реализовать модули, взаимодействующие с контроллерами Myrinet, Infiniband и Ethernet. Базовой функцией сетевого модуля является передача служебных сообщений и массивов данных через коммуникационную среду.

Рассмотрим процесс разработки стека RapidIO от драйвера контроллера RapidIO до реализации сетевого модуля для Nemesis и обоснуем принятые решения.

Драйвер контроллера RapidIO

Контроллер RapidIO в 1890BM6Я реализует следующие механизмы передачи данных стандарта RapidIO: IO-транзакции [10], сообщения и так

называемые дверные звонки (doorbell) [11]. Дверные звонки могут передать максимум 2 байта данных за одну транзакцию. Они предназначены, например, для оповещения о случившемся прерывании, а не для передачи массивов данных.

Использование IO-транзакций (в частности, NREAD и NWRITE – чтение и запись памяти удаленного узла) возможно путем выполнения процессорных инструкций чтения/записи специальной транслируемой области памяти, которая указывает на память удаленного узла. Каждая такая инструкция может либо загрузить, либо записать данные размером до 8 байт. Если учесть размер служебного заголовка для транзакции (6 байт + 6 бит), то он получается почти равным размеру передаваемых данных, что дает весьма низкую эффективность использования такого рода транзакций в данной реализации контроллера RapidIO.

Сообщения позволяют передавать до 4 Кб данных. Каждое сообщение состоит из сегментов, количество которых может достигать 16, а размер каждого из них не превышает 256 байт. По спецификации RapidIO гарантируются доставка сообщений до места назначения и сохранение порядка передачи с помощью полей «номер письма» (номер сообщения) и «номер сегмента» в заголовке. Оно также имеет дополнительное поле (подкраску) – номер почтового ящика, который служит для разделения входящих потоков сообщений. Таким образом, сообщения являются наиболее приемлемым способом передачи данных ввиду своей высокой эффективности и необходимости наименьшего числа транзакций для данных размером более 8 байт. Далее по тексту под буфером будет пониматься область памяти, разбитая на сообщения либо на отправку, либо на прием.

Контроллер сообщений RapidIO микросхемы 1890BM6Я имеет ряд особенностей.

- 8 выделенных очередей, программируемых на прием сообщений от заданного узла с заданным номером почтового ящика. С помощью механизма выделенных очередей становится возможным прием данных напрямую в буфер назначения без промежуточных копирований данных (zero-copy). Такой прием назовем приемом ожидаемого пакета.

- Наличие общей очереди, куда попадают сообщения, не попавшие ни в одну из выделенных очередей. Такие входящие сообщения будем называть неожиданными.

- Проблема сегментов разной длины. По спецификации RapidIO сообщение состоит максимум из 16 сегментов, размер которых одинаков (размер сегмента не больше 256 байт), кроме последнего сегмента. Отправитель посылает сегменты один за другим, одновременно ожидая статуса доставки каждого сегмента от узла получателя. При возникновении заторов в коммуникационной среде любой сегмент может быть отклонен получателем, что приводит к повторной отправке недоставленного

сегмента. Отсюда возможна следующая ситуация, когда все сегменты, кроме последнего, были отклонены получателем. Тогда этот успешно полученный последний сегмент расположится в памяти по адресу $\text{адрес_буфера} + 256 \text{ байт} * (\text{количество_сегментов} - 1)$. Если же размер всех предыдущих сегментов меньше 256 байт, можно с уверенностью сказать, что произошла порча памяти у получателя в момент записи последнего сегмента, если, конечно, буфер был рассчитан только на прием действительного размера сообщения. В качестве простого решения этой проблемы введем ограничение выравнивания размера буферов отправки и приема на 256 байт.

- Физический адрес буфера должен быть выравнен на 8 байт. Из предыдущего требования вытекает требование выравнивания адреса буфера, лежащего в транслируемой виртуальной памяти, на 256 байт. Но логика работы контроллера RapidIO микросхемы 1890BM6Я при приеме в выделенную очередь заключается в том, что количество отправленных сообщений и размер каждого из них должны точно совпадать с числом и размерами запрограммированных сообщений на стороне получателя. Поэтому для упрощения увеличим требование выравнивания для адреса такого буфера до 4Кб.

Для передачи сообщения на уровне прикладной программы, использующей MPI, на первом узле необходимо вызвать функцию передачи MPI_Send, а на другом – функцию приема MPI_Recv. Для MPI_Send задаются параметры: ID получателя, TAG сообщения («подкраска» сообщения), адрес буфера и его размер. Для MPI_Recv должны быть заданы соответствующие параметры, которые либо точно совпадают с параметрами отправки ID и TAG, либо представляют собой особые значения MPI_ANY_SOURCE для ID или MPI_ANY_TAG для TAG, которые пропускают любые соответствующие значения. Такой парный подход вызовов MPI_Send/MPI_Recv для приема/передачи данных позволяет воспользоваться выделенными очередями контроллера RapidIO микросхемы 1890BM6Я для приема и последующей отправки в них данных «без копирования». После успешного открытия выделенной очереди информация о номере почтового ящика должна быть передана узлу-отправителю. Для передачи такого рода контрольной информации зарезервируем нулевой почтовый ящик, а остальные номера предоставим выделенным очередям.

Таким образом, можно обозначить три интерфейсные функции нижнего уровня драйвера: отправка пакета (номер почтового ящика передается в параметрах запроса), прием неожиданных сообщений (контрольной информации) и запрос на прием ожидаемых пакетов (запрос на открытие выделенной очереди). Все эти функции должны рабо-

тать с допустимыми (то есть обладающими необходимыми требованиями выравнивания, введенными выше) для работы контроллера RapidIO буферами. Для обработки неожиданных сообщений драйвер при инициализации открывает общую очередь.

Управляющий уровень драйвера контроллера RapidIO

Так как число выделенных очередей и почтовых ящиков ограничено, а порядок вызовов MPI функций-запросов на прием, которые могут побудить открытие выделенной очереди, строго говоря, не детерминирован, необходимо осуществлять динамическое присвоение номера почтовому ящику и открытие выделенной очереди для одного определенного узла. Эта работа отнесена на отдельный управляющий уровень. При недостатке свободных очередей или почтовых ящиков запросы на выделенную очередь откладываются до тех пор, пока не станет доступным недостающая часть.

Для приема или передачи буфера без копирования необходимо, чтобы его физический адрес и размер удовлетворяли введенным требованиям выравнивания. При этом, если буфер находится в транслируемой виртуальной области памяти, осуществляется преобразование виртуального адреса в физические адреса, при необходимости с подкачкой страниц памяти с внешнего носителя. В случае неудовлетворения введенным требованиям выделяется «правильный» связующий буфер (bounce buffer) для обмена данными (либо на прием, либо на передачу) с применением копирования.

Прием в выделенную очередь (прием ожидаемого пакета) происходит только после открытия выделенной очереди и получения контрольной информации о выделенном номере почтового ящика отправителем. Но выделение почтового ящика (которого для данного узла в данный момент может и не быть), открытие выделенной очереди на прием (на данный момент свободной может и не быть) и отправка контрольной информации узлу-отправителю могут занять слишком много времени по отношению к размеру передаваемых данных. Поэтому для данных маленького размера дешевле обойтись общей очередью (нулевой почтовый ящик), используя связующий буфер. Чтобы отличить маленькие данные от контрольной информации, необходимо передавать заголовок с типом данных и их размером вместе с предполагаемыми данными.

Граница между маленькими и большими данными сильно зависит от самой аппаратуры, поэтому устанавливается эмпирическим путем. Если по результатам измерения эта граница располагалась за пределами 4 Кб, то данные маленького размера, превышающие 4 Кб, будут разбиты на не-

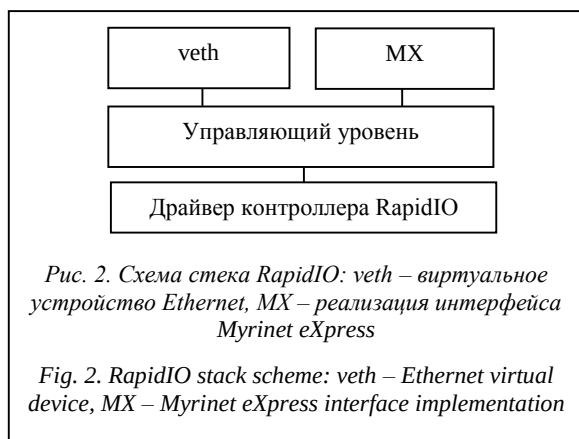
сколько сообщений RapidIO драйвером контроллера. Для последующей обработки таких данных на узле-получателе необходимо их все собрать в единый пакет, что возможно благодаря введенному заголовку.

Таким образом, в управляющем уровне можно выделить следующие интерфейсные функции: отправка данных в общую очередь узла-назначения (маленький размер данных или контрольная информация вместе с заголовком о типе данных и их размере – протокол управляющего уровня); прием неожиданных пакетов, собранных из нескольких сообщений; регистрация буфера (выделение при необходимости связующего буфера или получение физических адресов из транслируемых виртуальных); запрос на открытие выделенной очереди для приема в зарегистрированный буфер; отправка зарегистрированного буфера в заданный почтовый ящик.

В общем случае пользователей управляющего интерфейса может быть несколько. Основная проблема заключается в распознавании пользователя-получателя неожиданных пакетов, так как для ожидаемых пакетов получатель известен по определению. Для ее решения в заголовке неожиданного пакета необходимо передавать идентификатор пользователя-получателя. Для этого введем еще одну интерфейсную функцию-регистрацию пользователя-получателя, которая формирует рабочий контекст, и постулируем работу функций управляющего уровня в рамках этого контекста.

Высшие уровни и тестирование

В числе пользователей канала RapidIO – виртуальное устройство Ethernet (см. рис. 2). Стек Ethernet является базовой необходимостью, так как с его помощью работают многие полезные протоколы, например telnet и ssh. Другим пользователем является уровень MX (см. рис. 2), который реализует интерфейс передачи сообщений по значениям «ключ-маска», в достаточно хорошей степени точности соответствующий интерфейсу передачи сообщений MPI, описанному выше.



Интерфейс Myrinet eXpress [12], использующийся в сетевом модуле mx библиотеки MPICH, также соответствует интерфейсу передачи сообщений MPI. Этот сетевой модуль был взят как справочник при реализации библиотеки пользовательского уровня MXlib, которая реализует интерфейс Myrinet eXpress для работы с ядерным уровнем MX. С этой библиотекой стала возможной работа библиотеки MPICH поверх канала RapidIO через использование сетевого модуля mx.

В качестве аппаратуры для измерения производительности MPI были взяты 4 ПЛИС с ядром микросхемы семейства 1890BM6Я, соединенные каналами RapidIO, пиковая пропускная способность которого составляет 192 МБ/с. На одном, двух и четырех узлах были запущены тесты NAS Parallel Benchmarks (NPB) [13] класса A, которые достигли 80 %-ной эффективности масштабирования при увеличении количества узлов. Также была измерена производительность между двумя узлами тестами OSU Micro-Benchmarks [14], результаты которых представлены в таблице.

Результаты измерения производительности стека RapidIO при работе связки MX + MPICH

Measurement results for RapidIO stack performance with operating MX + MPICH

Размер передаваемых данных (байт)	Тест		
	osu_bibw (МБ/с)	osu_bw (МБ/с)	osu_latency (мкс)
1	0,00	0,00	699,62
64	0,23	0,19	639,04
512	1,78	1,51	680,80
2048	6,52	5,49	804,26
4096	10,55	8,44	953,26
16384	24,32	18,34	2194,66
65536	59,35	49,56	3906,18
131072	72,78	67,65	7441,22
524288	112,04	81,08	31151,76
4194304	219,03	162,00	219750,74

На основании изложенного можно сделать следующие выводы. Реализованный в данной статье стек RapidIO для MPI показал свою работоспособность на различных программах – тесты NAS Parallel Benchmarks, OSU Micro-Benchmarks, тесты MPICH. На тесте передачи сообщений OSU удалось достичь скорости передачи в 84 % от пиковой. Благодаря расширению области тестирования путем запуска разнообразных MPI-программ были выявлены и устранены ранее неизвестные ошибки в логике работы контроллера RapidIO.

При использовании готовых библиотек добиться максимальной производительности можно только на аппаратуре, функциональные возможности которой соответствуют идеологии обмена данными, заложенной в самой библиотеке. В целом контроллер RapidIO микросхемы 1890BM6Я лишь частично соответствует стандарту MPI, а соответственно, и его библиотечным реализациям. Так,

например, для передачи большого массива данных необходимо произвести «ручное» программирование каждого сообщения в соответствии с таблицей физических адресов, в то время как такая тривиальная работа может быть реализована в железе. Необходимость разрешения конфликтов выравнивания буферов на обоих узлах также влечет дополнительные накладные расходы на процессор, в худшем случае прибегая к копированию данных.

Другим примером является введенный в стандарт MPI версии 2 интерфейс RMA (Remote Memory Access), который позволяет совершать операции чтения и записи памяти любого размера удаленного узла. Микросхема 1890VM6Я предоставляет весьма скудный механизм такого удаленного доступа к памяти – IO-транзакции типа NREAD и NWRITE с полезной нагрузкой до 8 байт, хотя по стандарту RapidIO полезная нагрузка может составлять 256 байт.

Недостатком является и получившаяся сильная перегруженность стека драйверов, что делает отправку и прием данных слишком дорогой операцией. По грубым оценкам, получилось больше 20 тысяч процессорных инструкций для передачи всего лишь одного байта данных. Это означает, что при частоте процессора в 1 ГГц передать один байт получится не меньше, чем через несколько десятков микросекунд. Конечно, с ростом передаваемого объема данных сама подготовка к передаче будет становиться исчезающе малой, но факт задержки, тем не менее, останется. Поэтому необходима дальнейшая работа по ее уменьшению.

Автор выражает благодарность Лазутину Юрию Михайловичу (ФНЦ НИИСИ РАН) за помощь в подготовке данной статьи.

Литература

1. Message Passing Interface Forum. MPI: A message-passing interface standard, version 3.0, 2012. URL: <http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf> (дата обращения: 25.07.2015).
2. Бобков С.Г., Еремин А.А., Кондратьева Н.В., Сердин О.В. Разработка высоконадежных многопроцессорных модулей на базе высокоскоростных каналов RapidIO // Программные продукты и системы. 2013. № 4. С. 49–55.
3. Yang C., Xiao L., Lu Yu., Liao X. MilkyWay-2 supercomputer: system and application. Front. Comput. Sci., 2014, no. 8 (3), pp. 345–356.
4. Graham R.L., Gutierrez S.K., Hjelm N.T., Venkata M.G. Performance Evaluation of Open MPI on Cray XE/XK Systems. High-Performance Interconnects (HOTI), 2012 IEEE 20th Annual Symposium, 22–24 Aug. 2012, pp. 40–47.
5. Sato M., Tsuji M. / Performance Evaluation of OpenMP and MPI Hybrid Programs on a Large Scale Multi-core Multi-socket Cluster, T2K Open Supercomputer. Parallel Processing Workshops, 2009. ICPPW '09. Intern. Conf., 22–25 Sept. 2009, pp. 206–213.
6. MPICH is a high performance and widely portable implementation of the Message Passing Interface (MPI) standard. URL: <https://www.mpich.org/> (дата обращения: 25.07.2015).
7. Архитектура библиотеки MPICH первой версии. URL: <http://www.mcs.anl.gov/research/projects/mpi/mpich1-old/papers/mpicharticle/node21.html#Node21> (дата обращения: 25.07.2015).
8. Pritchard H., Gorodetsky I., Buntinas D. A uGNI-Based MPICH2 Nemesis Network Module for Cray XE Computer Systems. EuroMPI'11 Proceedings of the 18th European MPI Users' Group conference on Recent advances in the message passing interface, pp. 110–119.
9. Buntinas D., Mercier G., and Gropp W. Design and Evaluation of Nemesis, a Scalable, Low-Latency, Message-Passing Communication Subsystem. In CCGRID'06, 2006, pp. 521–530.
10. RapidIO Interconnect Specification. Part 1: Input/Output Logical Specification. Revision 2.2. RapidIO Trade Association, 2011. URL: http://www.rapidio.org/specs/current/RapidIO_Rev_2.2_Specification.zip (дата обращения: 25.07.2015).
11. RapidIO Interconnect Specification. Part 2: Message Passing Logical Specification. Revision 2.2. RapidIO Trade Association, 2011. URL: http://www.rapidio.org/specs/current/RapidIO_Rev_2.2_Specification.zip (дата обращения: 25.07.2015).
12. Myricom, Inc. Myrinet Express (MX): A High Performance, Low-Level, Message-Passing Interface for Myrinet. URL: <http://www.myri.com/scs/MX/doc/mx.pdf> (дата обращения: 25.07.2015).
13. NAS Parallel Benchmarks: небольшой набор программ, созданных для помощи в оценке производительности параллельных суперкомпьютеров. URL: <http://www.nas.nasa.gov/publications/npb.html> (дата обращения: 25.07.2015).
14. OSU Micro-Benchmarks: тесты производительности вычислительных узлов с использованием MPI. URL: <http://mvapich.cse.ohio-state.edu/benchmarks/> (дата обращения: 25.07.2015).

DOI: 10.15827/0236-235X.112.093-098

Received 27.07.15

MPI SUPPORT IN LINUX OS FOR RapidIO-BASED MULTIPROCESSOR COMPUTER SYSTEMS

Kuleshov A.S., Postgraduate Student, rndfax@cs.niisi.ras.ru
(SRISA RAS, Nakhimovsky Av. 36/1, Moscow, 117218, Russia Federation)

Abstract. Solving calculus problems using multiprocessor computing systems refers to a certain data transfer programming interface. The most well-known interface is the message passing interface (MPI), which defines the standard of API functions to transfer data between nodes for end-user programs. This paper presents MPI implementation for SoC 1890VM6IA developed in SRISA RAS. It is a base for multipurpose multiprocessor computing systems, nodes of which can be connected through high speed RapidIO channels. To implement it we took MPICH, which is the most widespread library of MPI, and Linux kernel that proved itself as the best solution for supercomputing. MPICH basic configuration includes different network modules, which implement API for specific transport drivers: tcp (Ethernet TCP/IP), ib (Infiniband), mx (Myrinet eXpress) and others. Using network module mx, which is well-fitting for RapidIO architecture, gave the opportunity to focus only on the development of the Linux driver for RapidIO controller. The features of RapidIO controller of 1890VM6IA, MPI interface and the idea of using RapidIO channel for purposes other than MPI in Linux, allowed implementing a message passing stack with the some degree of universality using RapidIO channel without unnecessary data copying. In the end of the paper there are the results of different MPI test programs

such as NAS Parallel Benchmarks and OSU Micro-Benchmarks for 4 nodes using the RapidIO channels. The author made a conclusion and summarized the results using RapidIO controller of 1890VM61A SoC as a transport facility for MPI.

Keywords: RapidIO, MPI, MPICH, Linux.

References

1. *Message Passing Interface Forum. MPI: A message-passing interface standard, version 3.0*. 2012. Available at: <http://www.mpi-forum.org/docs/mpi-3.0/mpi30-report.pdf> (accessed July 25, 2015).
2. Bobkov S.G., Eremin A.A., Kondratyeva N.V., Serdin O.V. Design of high reliability multiprocessor modules based on high-performance RapidIO interconnect architecture. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 4, pp. 49–55 (in Russ.).
3. Yang C., Xiao L., Lu Yu., Liao X. MilkyWay-2 supercomputer: system and application. *Front. Comput. Sci.* 2014, no. 8 (3), pp. 345–356.
4. Graham R.L., Gutierrez S.K., Hjelm N.T., Venkata M.G. Performance Evaluation of Open MPI on Cray XE/XK Systems. *IEEE 20th Annual Symp. on High-Performance Interconnects (HOTI)*. 2012, pp. 40–47.
5. Sato M., Tsuji M. Performance Evaluation of OpenMP and MPI Hybrid Programs on a Large Scale Multi-core Multi-socket Cluster, T2K Open Supercomputer. *ICPPW '09. Int. Conf. on Parallel Processing Workshops*. 2009, pp. 206–213.
6. *MPICH is a high performance and widely portable implementation of the Message Passing Interface (MPI) standard*. Available at: <https://www.mpich.org> (accessed July 25, 2015).
7. *Arkhitektura biblioteki MPICH pervoy versii* [MPICH library architecture, the 1st version]. Available at: <http://www.mcs.anl.gov/research/projects/mpi/mpich1-old/papers/mpicharticle/node21.html#Node21> (accessed July 25, 2015).
8. Pritchard H., Gorodetsky I. A uGNI-Based MPICH2 Nemesis Network Module for Cray XE Computer Systems. *Proc. of the 18th European MPI Users' Group Conf. on Recent Advances in the Message Passing Interface*. 2011, pp. 110–119.
9. Buntinas D., Mercier G., Gropp W. Design and Evaluation of Nemesis, a Scalable, Low-Latency, Message-Passing Communication Subsystem. *CCGRID '06*. 2006, pp. 521–530.
10. *RapidIO Interconnect Specification. Part 1: Input/Output Logical Specification. Revision 2.2*. RapidIO Trade Association. 2011. Available at: http://www.rapidio.org/specs/current/Ra-pidIO_Rev_2.2_Specification.zip (accessed July 25, 2015).
11. *RapidIO Interconnect Specification. Part 2: Message Passing Logical Specification. Revision 2.2*. RapidIO Trade Association. 2011. Available at: http://www.rapidio.org/specs/current/Ra-pidIO_Rev_2.2_Specification.zip (accessed July 25, 2015).
12. *Myricom, Inc. Myrinet Express (MX): A High Performance, Low-Level, Message-Passing Interface for Myrinet*. Available at: <http://www.myri.com/scs/MX/doc/mx.pdf> (accessed July 25, 2015).
13. *NAS Parallel Benchmarks*. Available at: <http://www.nas.nasa.gov/publications/npb.html> (accessed July 25, 2015).
14. *MVAPICH: MPI over InfiniBand, 10GigE/iWARP and RoCE*. Available at: <http://mvapich.cse.ohio-state.edu/benchmarks/> (accessed July 25, 2015).

Примеры оформления статьи в списке литературы

1. Кулешов А.С. Поддержка протокола MPI в ядре ОС Linux для многопроцессорных вычислительных комплексов на базе высокоскоростных каналов RapidIO // Программные продукты и системы. 2015. № 4 (112). С. 93–98.

2. Кулешов А.С. Поддержка протокола MPI в ядре ОС Linux для многопроцессорных вычислительных комплексов на базе высокоскоростных каналов RapidIO // Программные продукты и системы. 2015; DOI: 10.15827/0236-235X.112.093-098.

3. Kuleshov A.S. MPI support in Linux OS for RapidIO-based multiprocessor computer systems. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 4 (112), pp. 93–98 (in Russ.); DOI: 10.15827/0236-235X.112.093-098.

УДК 004.946

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.099-104

СИСТЕМА ВИЗУАЛИЗАЦИИ ТЕКСТУРИРОВАННЫХ МОДЕЛЕЙ ПЛАНЕТ ДЛЯ ТРЕНИРОВОК ПРОВЕДЕНИЯ КОСМИЧЕСКИХ ЭКСПЕРИМЕНТОВ

(Исследования выполнены при поддержке РФФИ, грант № 14-07-31332)

П.Ю. Тимохин, научный сотрудник, webpismo@yahoo.de
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)

Современным инструментом для проведения тренировок космических экспериментов, связанных с наблюдением планеты с околопланетной орбиты, является визуализация в масштабе реального времени виртуальной трехмерной модели планеты с детализированными текстурами, созданными на основе спутниковых снимков. Исследования показывают, что размеры и объем необходимых текстур существенно превышают аппаратно допустимые значения современных графических ускорителей (сверхбольшие текстуры), что не позволяет загружать их целиком в видеопамять и аппаратно обрабатывать. В статье описываются технология и программная реализация, позволяющие визуализировать в масштабе реального времени сверхбольшие текстуры планет, разбитые на небольшие участки одинакового размера (страницы). Ключевая идея предложенной технологии состоит в сокращении временных затрат на отбор видимых страниц сверхбольшой текстуры путем распараллеливания вычислений на графическом процессоре (GPU) с помощью ряда разработанных шейдерных программ. Технология обеспечивает текстурную детализацию для моделируемой поверхности планеты, наиболее близкую к разрешению экрана, а также позволяет визуализировать на виртуальной поверхности планеты в масштабе реального времени сверхбольшие текстуры, имеющие различное исходное разрешение.

Предложенная технология реализована в программном комплексе визуализации детализированных текстур планет, который интегрируется во внешнее приложение. Созданный программный комплекс также снабжен модулем расчета орбитального положения наблюдателя и пользовательским интерфейсом для управления полетом наблюдателя. В работе выполнена апробация реализованного программного комплекса в составе системы визуализации трехмерных виртуальных сцен, которая показала его адекватность поставленным задачам. Разработанный программный комплекс может быть использован как для улучшения характеристик существующих тренажерных систем по проведению космических экспериментов, так и для построения новых перспективных тренажеров.

Ключевые слова: визуализация, космический эксперимент, виртуальная планета, сверхбольшая текстура, реальное время.

Космические эксперименты, связанные с наблюдением планеты с околопланетной орбиты, являются источником ценных и уникальных научных данных о глобальных процессах и явлениях, протекающих на ее поверхности. В настоящее время выполнение таких экспериментов по исследованию поверхности Земли – одна из важных задач экипажей *Международной космической станции* (МКС). Мониторинг земной поверхности осуществляется через иллюминаторы как невооруженным глазом, так и с помощью фото- и видеоаппаратуры с изменяемой кратностью увеличения и с постоянным полем зрения [1]. Ввиду высокой сложности выполнения таких экспериментов (время эффективного наблюдения объекта ограничено 50–80 секундами) для получения успешного результата космонавту необходимо активно тренировать соответствующие навыки.

Современным инструментом для проведения таких тренировок является визуализация в масштабе реального времени виртуальной трехмерной модели планеты с детализированными текстурами, созданными на основе спутниковых снимков. Исследования в работе [1] показывают, что для имитации наблюдения Земли с борта МКС невооруженным глазом детализация таких текстур должна составлять не менее 150 м/пиксел и быть на поряд-

ков выше для имитации работы средств наблюдения с изменяемой кратностью увеличения. Проблема состоит в том, что размеры и объем таких текстур существенно превышают аппаратно допустимые значения современных графических ускорителей (сверхбольшие текстуры), что не позволяет загружать их целиком в видеопамять и аппаратно обрабатывать. Задача еще более усложняется в случае необходимости визуализации в космическом эксперименте дополнительных детализированных текстурных слоев (например, текстур с изображением облачного покрова или огней ночных городов). Все это обуславливает высокую востребованность технологий и систем визуализации, которые могут работать с огромными массивами текстурных данных в режиме реального времени.

Возможность наблюдения текстурированных виртуальных поверхностей планет с орбитальной высоты предоставляют ряд популярных виртуальных 3D-глобусов, разрабатываемых как крупными организациями, так и сообществами энтузиастов. Примерами таких программных продуктов являются виртуальный географический атлас NASA World Wind [2], астрономическая программа Celestia [3], симулятор орбитальных космических полетов Orbiter [4] и другие. Однако возможности и скорость визуализации детализированных тек-

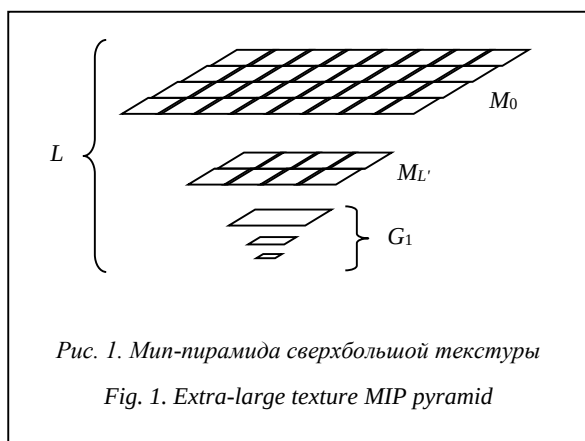
стур планет в подобных системах зачастую ограничены и не отвечают в полной мере требованиям для постановки космических экспериментов [1]. К тому же особенности подобных зарубежных программных продуктов затрудняют их применение в отечественной космической отрасли.

Для решения рассматриваемой задачи в данной работе предлагается разработанная в НИИСИ РАН система визуализации текстурированных моделей планет в масштабе реального времени. Принцип действия предлагаемой системы основан на разработанной технологии визуализации сверхбольших текстур.

Технология визуализации сверхбольших текстур

В основе предлагаемой технологии лежит подход, применяемый в компьютерной графике для визуализации текстур, размеры и объем которых удовлетворяют аппаратным возможностям современных видеокарт. При данном подходе наряду с исходными текстурами применяются текстуры с меньшими уровнями детализации (мип-пирамида текстур), причем для каждого пиксела синтезируемого изображения подбираются текстуры тех уровней, детализация которых наиболее близка к разрешению экрана (технология аппаратного мипмэппинга [5]).

В предлагаемой технологии по исходной сверхбольшой текстуре (далее текстура T) также создается мип-пирамида текстур из L уровней, причем текстуры уровней от 0 до L' разбиваются на небольшие участки одинакового размера (страницы), образуя массивы $M_0, \dots, M_{L'}$ страниц (рис. 1), которые размещаются на жестком диске в сжатом виде. Перед визуализацией выделяется область видеопамети, в которую на постоянное хранение помещаются текстуры уровней от $L'+1$ до $L-1$ (текстурная группа G_1), а также области видео- и оперативной памяти, куда будут динамически отбираться и подкачиваться страницы из массивов $M_0, \dots, M_{L'}$ (текстурные группы G_2 и B), необходимые для визуализации текущего кадра.



Ввиду того, что избыточная загрузка в видеопаметь страниц препятствует соблюдению режима реального времени визуализации (работа с жестким диском является потенциально узким местом), а отсутствие необходимых страниц существенно отражается на качестве синтезируемого изображения планеты, в предлагаемой технологии акцент сделан на выполнении точного отбора видимых страниц по всем пикселям кадра. При современных разрешениях средств отображения (Full HD, Ultra HD) объем пикселей, которые необходимо обрабатывать, крайне велик, поэтому ключевая идея предлагаемой технологии состоит в сокращении временных затрат путем распараллеливания такой обработки полностью на графическом процессоре (GPU).

По подобному направлению пошли авторы работы [6], однако предложенное ими решение на базе программно-аппаратной архитектуры CUDA позволяет обрабатывать в реальном времени лишь некоторую выборку из всех пикселей кадра и ограничивает область средств реализации только видеокартами компании NVidia. В отличие от [6] в данной работе предлагается распараллеливать обработку всех пикселей кадра с помощью ряда разработанных шейдерных программ, написанных на универсальном языке GLSL, поддерживаемом всеми современными графическими ускорителями. Предлагаемая технология включает в себя ряд этапов.

Синтез экранной карты текстурных координат. На данном этапе вычисляется двухканальная текстура, имеющая размеры кадра (полнокадровая текстура), в каждый тексел которой записываются текстурные координаты (s, t) видимой точки модели поверхности планеты (далее ST-текстура). С этой целью при создании трехмерной модели поверхности планеты для каждой ее геометрической вершины указываются текстурные координаты этой вершины в исходной сверхбольшой текстуре T . Используя эти координаты, при синтезе ST-текстуры планеты для каждого пиксела видеокарта автоматически вычисляет текстурные координаты (s, t) на текстуре T , соответствующие этому пикселу. Непосредственно сам синтез ST-текстуры реализуется путем рендеринга модели планеты в текстуру с помощью разработанной шейдерной программы.

Синтез экранной карты значений уровня текстурной детализации. На данном этапе вычисляется одноканальная полнокадровая текстура, в каждый тексел которой записывается уровень ℓ текстурной детализации видимой точки модели поверхности планеты (далее LOD-текстура). Данная величина является вещественным числом и определяется на основе вычисления количества текселов, охватываемых сторонами проекции пиксела в пространстве текстуры T . На практике такая проекция эффективно приближается параллело-

граммом, длины сторон которого вычисляются с помощью частных производных текстурных координат (s, t) по экранным координатам пикселей. Более подробно вычисление ℓ рассмотрено в [7]. В разработанной шейдерной программе, реализующей вычисление ℓ , необходимые текстурные координаты (s, t) считываются из тексела ST-текстуры, имеющего такое же расположение, что и рассчитываемый текстел LOD-текстуры.

Получение списка видимых страниц. Формирование списка страниц, видимых в текущем кадре, осуществляется с помощью вычисленных выше ST-текстуры и LOD-текстуры и выполняется в 3 шага.

На 1-м шаге создается полнокадровая текстура U_1 , в каждый текстел которой записывается идентификатор видимой страницы. Этот идентификатор задается тройкой чисел (q, i, j) , где q – наименьший номер уровня мип-пирамиды, который необходимо использовать для закраски текущего пиксела, а i и j – номера строки и столбца страницы в массиве M_0 страниц:

$$q = \min(\max(0, \lfloor \ell \rfloor), L' + 1),$$

$$i = \min(\lfloor h_q \cdot t \rfloor, h_q - 1) \cdot 2^q,$$

$$j = \min(\lfloor w_q \cdot s \rfloor, w_q - 1) \cdot 2^q,$$

где ℓ – величина уровня детализации текстуры T , записанная для текущего пиксела в ST-текстуре; h_q , w_q – число строк и столбцов массива M_q страниц; (s, t) – текстурные координаты в пространстве текстуры T , записанные для текущего пиксела в LOD-текстуре.

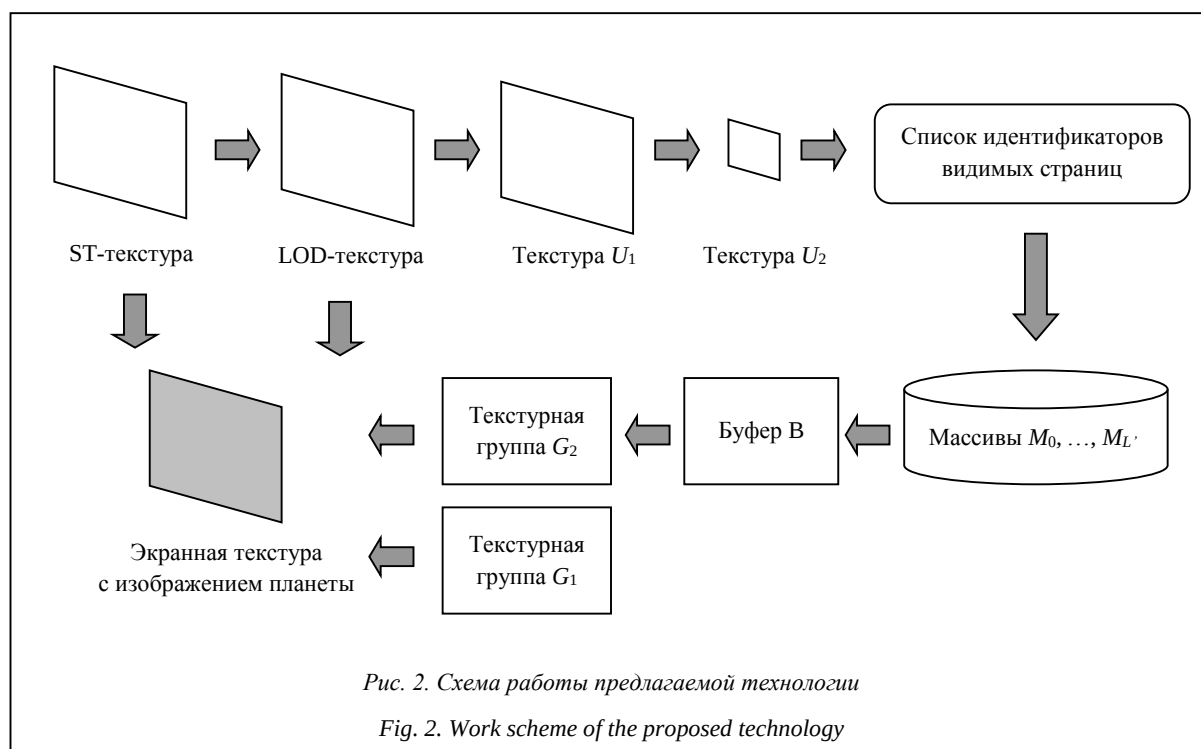
На 2-м шаге осуществляется удаление повторяющихся идентификаторов в текстуре U_1 . Для этого сначала формируется множество V двумерных геометрических вершин, в которое для каждого идентификатора (q, i, j) из U_1 записывается вершина $V_{q,i,j}$ с координатами $x = -1 + 2j/w_0$, $y = -1 + 2i/h_0$ и цветом 2^q . Затем с помощью специально разработанного геометрического шейдера множество V параллельно визуализируется в одноканальную текстуру U_2 размера $w_0 \times h_0$ с включенной побитовой операцией «ИЛИ» записи цвета. В результате такой визуализации в текселах текстуры U_2 будут записаны битовые векторы, в которых единицы стоят в тех позициях, номера которых соответствуют номерам видимых страниц из массивов $M_0, \dots, M_{L'}$.

На 3-м шаге формируется список неповторяющихся идентификаторов страниц. С помощью специально разработанного геометрического шейдера создается множество V' трехмерных вершин, в которые записываются идентификаторы страниц, отмеченных в текстуре U_2 единичными битами. Затем выполняется рендеринг вершин из V' в буфер вершин. В результате в буфере вершин формируется список неповторяющихся идентификаторов отобранных страниц.

Подкачка видимых страниц. Так как графические процессоры могут работать только с информацией, находящейся в видеопамяти, а в видеопамять информацию можно копировать только из оперативной памяти, в работе предлагается использовать массивы $M_0, \dots, M_{L'}$ страниц на жестком диске, массив B в оперативной памяти и массив G_2 в видеопамяти. Последние два массива используются для кэширования. То есть при необходимости использования какой-либо страницы она сначала ищется в массиве G_2 . Если ее там нет, то она подкачивается из массива B , а если ее нет и в этом массиве, она подкачивается с жесткого диска. При этом новая загружаемая страница записывается на место той, которая дольше всех не использовалась (стратегия LRU, Least Recently Used). Подкачка страниц из $M_0, \dots, M_{L'}$ в B выполняется непрерывно в отдельном потоке. Для подкачки страниц из B в G_2 отводится некоторый промежуток времени, по истечении которого подкачка прекращается, а загрузка оставшихся страниц переносится в следующий кадр визуализации.

Синтез экранной текстуры с изображением планеты. Закраска пикселей синтезируемой текстуры с изображением планеты выполняется с помощью текстурных групп G_1 и G_2 , а также ST-текстуры и LOD-текстуры. Способ вычисления цвета закраски для каждого пиксела определяется индивидуально, исходя из его величины ℓ текстурной детализации, считанной из LOD-текстуры. Из этой величины вычисляются наименьший номер $q = \lfloor \ell \rfloor$ уровня мип-пирамиды, а также коэффициент $f = \{ \ell \}$ интерполяции между q -м и $q+1$ -м уровнями мип-пирамиды. В случае $q > L'$ цвет пиксела вычисляется путем аппаратной выборки из текстурной группы G_1 по текстурным координатам (s, t) пиксела, считанным из ST-текстуры. Если $q = L'$, то цвет вычисляется путем интерполяции с коэффициентом f двух цветов, один из которых вычисляется аналогично первому случаю, а второй – путем выборки из соответствующей координатам (s, t) страницы q -го уровня, загруженной в массив G_2 . Если $q \in [0, L'-1]$, цвет пиксела вычисляется также путем интерполяции с коэффициентом f двух цветов, но вычисленных уже путем выборки из страниц q -го и $q+1$ -го уровней. Если необходимая пара страниц отсутствует в массиве G_2 , для визуализации используется уже подкаченная пара страниц ближайшего более грубого уровня детализации, а в случае отсутствия таковых – данные из постоянно хранимой группы G_1 текстур. Цвета всех закрасиваемых пикселей вычисляются параллельно на GPU с помощью разработанного фрагментного шейдера.

На рисунке 2 изображена результирующая схема работы предложенной технологии. Описанная технология легко применяется для визуализации нескольких сверхбольших текстур на поверхности планеты. Для каждой сверхбольшей тек-



туры создаются свои массивы $M_0, \dots, M_L$ страниц, буфер B и текстурные группы G_1 и G_2 . Этап вычисления ST-текстуры планеты выполняется один раз в кадре для всех визуализируемых текстур, а оставшиеся этапы технологии выполняются для каждой сверхбольшой текстуры отдельно. Синтезированные в результате экранные текстуры с различными изображениями планеты смешиваются в соответствии с типом данных, который они представляют (например, дневная поверхность и облачный покров), или с задачей, поставленной в космическом эксперименте.

Результаты исследования

Описанная технология реализована в программном комплексе визуализации текстурированных моделей планет. Программный комплекс выполнен в виде dll-библиотеки, которая интегрируется во внешнее приложение (например, в систему визуализации трехмерных виртуальных сцен [8]). Такой подход позволяет путем минимальных усилий построить любой программный комплекс, использующий высококачественную визуализацию сверхбольших текстур планет в режиме реального времени. В качестве базовой технологии для вычисления описанных в работе текстур и вывода трехмерной графики система использует графическую библиотеку OpenGL и язык программирования шейдеров GLSL, соответствующие стандарту OpenGL 3.3. Разработанный программный комплекс также снабжен модулем расчета орбитального положения наблюдателя, созданным на основе представленной ранее технологии моделиро-

вания полета космического аппарата по орбите [9, 10], и пользовательским интерфейсом, позволяющим задавать параметры эллиптической кеплеровской орбиты виртуального наблюдателя, начало и конец участка подспутниковой трассы и т.д. На рисунке 3 изображено диалоговое окно для

Рис. 3. Диалоговое окно для управления полетом виртуального наблюдателя по околоземной орбите

Fig. 3. Dialog box for controlling the virtual observer's flight along a near-Earth orbit

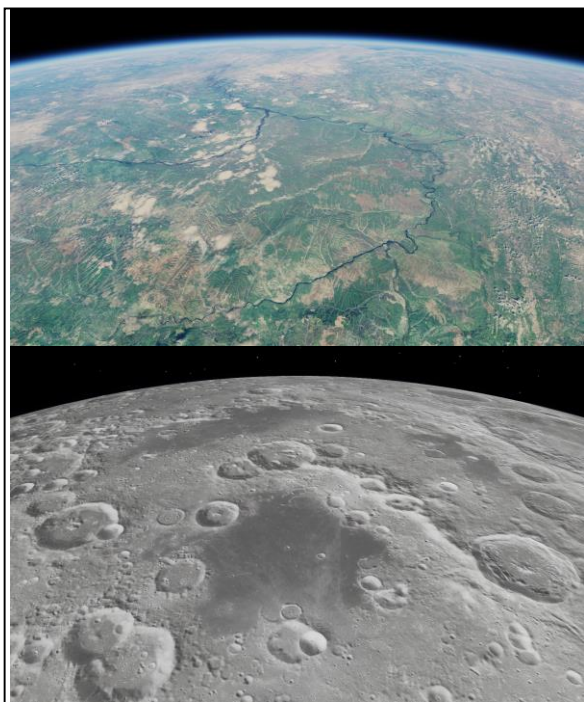


Рис. 4. Примеры изображений текстурированных моделей Земли и Луны, синтезированных в реальном времени с помощью разработанной системы визуализации

Fig. 4. Examples of images of the Earth and Moon textured models, rendered in real-time using developed visualization system

управления полетом виртуального наблюдателя вдоль участка подспутниковой трассы, соответствующего 1-, 2- и 3-му орбитальным виткам МКС.

Разработанный программный комплекс прошел апробацию в составе системы визуализации трехмерных виртуальных сцен GLView [8]. Для апробации использовались трехмерная полигональная модель земной и лунной поверхности (около 150 тыс. полигонов каждая), а также сверхбольшие текстуры с изображением подстилающей поверхности Земли, ее облачного покрова и Луны. Скорость визуализации текстурированной земной поверхности составила около 150 кадров в секунду, а лунной – около 200 кадров в секунду.

Апробация комплекса проводилась также в составе сложной виртуальной сцены с моделью МКС. Тестируемая сцена содержала трехмерную текстурированную полигональную модель МКС (более 2 млн полигонов), для которой выполнялся реалистичный расчет освещения и теней. Скорость визуализации такой сцены составила около 76 кадров/с.

Синтез всех изображений выполнялся на видеокарте типа GeForce GTX 680 в разрешении Full HD. На рисунке 4 изображены примеры кадров визуализации текстурированных моделей земной и лунной поверхности.

В заключение отметим, что в данной работе представлена технология визуализации сверхбольших текстур на виртуальных поверхностях планет в масштабе реального времени. Технология обеспечивает текстурную детализацию для моделируемой поверхности планеты, наиболее близкую к разрешению экрана, а также позволяет визуализировать на виртуальной поверхности планеты в масштабе реального времени несколько сверхбольших текстур, имеющих различное исходное разрешение. Предложенная технология реализована в программном комплексе визуализации детализированных текстур планет. Апробация комплекса показала его адекватность поставленным задачам. Он может быть использован как для улучшения характеристик существующих тренажеров по проведению космических экспериментов, так и для построения новых перспективных тренажеров, связанных, например, с исследованием Луны и Марса.

Литература

1. Митин А.И., Брагин В.И. Пути повышения адекватности моделирования визуальных условий мониторинга земной поверхности на тренажере служебного модуля Российского сегмента Международной космической станции // Пилотируемые полеты в космос. 2014. № 3. С. 60–70.
2. NASA World Wind. URL: <http://worldwind.arc.nasa.gov> (дата обращения: 10.07.2015).
3. Celestia. URL: <http://www.shatters.net/celestia> (дата обращения: 10.07.2015).
4. Orbiter. URL: <http://orbit.medphys.ucl.ac.uk> (дата обращения: 10.07.2015).
5. Ewins J. et al. MIP-Map Level Selection for Texture Mapping. IEEE Transactions on Visualization and Computer Graphics, 1998, vol. 4, no. 4, pp. 317–329.
6. Hollemeersch C.F., Pieters B., Lambert P., Van de Walle R. Accelerating virtual texturing using CUDA. GPU Pro: Advanced Rendering Techniques, 2010, vol. 1, pp. 623–641.
7. Тимохин П.Ю., Михайлюк М.В. Сверхбольшие текстуры для высокореалистичной визуализации виртуальных ландшафтов // Информационные технологии и вычислительные системы. 2013. № 3. С. 46–54.
8. Михайлюк М.В., Торгашев М.А. Система визуализации «GLView» для имитационно-тренажерных комплексов подготовки космонавтов // Пилотируемые полеты в космос. 2013. № 4. С. 60–74.
9. Тимохин П.Ю. Моделирование полета космического аппарата по околоземной орбите в космическом тренажерном комплексе // Программные продукты и системы. 2010. № 4. С. 36–41.
10. Михайлюк М.В., Тимохин П.Ю. Моделирование взаимного расположения планеты и спутника в космических видеотренажерах // Mathematica Montisnigri. 2014. Т. XXIX. С. 91–97.

Abstract. A modern tool to train space experiments related to the planet observation from a circumplanetary orbit is real-time visualization of virtual 3D planet model with detailed textures based on satellite images. Studies show that the size and memory space required for such textures significantly exceeds the capability of modern graphics accelerators (extra-large textures). This does not allow loading such textures in video memory completely and their hardware processing. This article describes a technology and software implementation which allows real-time visualizing of extra-large planet textures divided into small pieces of an equal size (pages). The key idea of the proposed technology is to reduce time costs for visible pages selection using GPU parallelizing computations by means of developed shader programs. The technology provides texture detailing of the simulated planet surface to fit the screen resolution. It also allows real-time visualizing multiple extra-large textures with different initial resolution on a virtual planet surface. The proposed technology is implemented in the planet texture visualization software package, which can be integrated in an external application. The created software package is also equipped with a module which calculates an observer's orbital position, as well as a user interface to control an observer's flight. The paper considers testing of the implemented software package as a part of 3D virtual scene visualization system. The results show that it meets all the requirements. The developed software package can be used both to improve features of existing simulators for space scientific experiments and to build new promising ones.

Keywords: visualization, space experiment, virtual planet, extra-large texture, real-time.

References

1. Mitin A.I., Bragin V.I. Ways of Improving the Adequacy of Modeling Visual Conditions of the Earth Surface Monitoring on the Service Module Simulator of the ISS RS. *Pilotiruemye polety v kosmos [Manned Spaceflight]*. 2014, no. 3, pp. 60–70 (in Russ.).
2. NASA World Wind. Available at: <http://worldwind.arc.nasa.gov> (accessed July 10, 2015).
3. Celestia. Available at: <http://www.shatters.net/celestia> (accessed July 10, 2015).
4. Orbiter. Available at: <http://orbit.medphys.ucl.ac.uk> (accessed July 10, 2015).
5. Ewins J., Waller M.D., White M., Lister P.F. MIP-Map Level Selection for Texture Mapping. *IEEE Transactions on Visualization and Computer Graphics*. 1998, vol. 4, no. 4, pp. 317–329.
6. Hollemeersch C.F., Pieters B., Lambert P., Van de Walle R. Accelerating virtual texturing using CUDA. *GPU Pro: Advanced Rendering Techniques*. 2010, vol. 1, pp. 623–641.
7. Timokhin P.Yu., Mikhaylyuk M.V. Extra-large textures for high realistic terrain visualization. *Informatsionnye tekhnologii i vychislitelnye sistemy [Information Technologies & Computing Systems]*. 2013, no. 3, pp. 46–54 (in Russ.).
8. Mikhaylyuk M.V., Torgashev M.A. “GLView” visualization system for cosmonaut preparing simulation and training complexes. *Pilotiruemye polety v kosmos [Manned Spaceflight]*. 2013, no. 4, pp. 60–74 (in Russ.).
9. Timokhin P.Yu. Near-earth orbit flight simulation in a space training system. *Programmye produkty i sistemy [Software & Systems]*. 2010, no. 4, pp. 36–41 (in Russ.).
10. Mikhaylyuk M.V., Timokhin P.Yu. Simulation of mutual planet-satellite arrangement in space video simulators. *Mathematica Montisnigri*. 2014, vol. XXIX, pp. 91–97 (in Russ.).

Реклама

НИИ «Центрпрограммсистем»

Предлагаем комплексные решения по созданию систем защиты информации для существующих и разрабатываемых автоматизированных (информационных) систем:

- системы защиты конфиденциальной информации;
- системы защиты персональных данных до уровня К1;
- системы защиты информации от несанкционированного доступа и иностранных технических разведок, позволяющие обрабатывать информацию, отнесенную к государственной тайне с грифом до «совершенно секретно» включительно;
- аппаратно-программные модули доверенной загрузки;
- аппаратно-программные комплексы защиты информации.

Разработка ведется строго в соответствии с требованиями российского законодательства, руководящих документов ФСТЭК России, ФСБ России, нормативных документов МО РФ и других ведомств.

Правомочность выполнения работ подтверждается действующими лицензиями ФСБ, ФСТЭК и Минобороны России.

Опыт работы НИИ «Центрпрограммсистем» в области информационной безопасности – более 20 лет.

Подробности на сайте www.szi-cps.ru
Телефон в Москве: (495) 729-25-84
Телефон в Твери: (4822) 68-06-36

УДК 004.921

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.105-109

ОПРЕДЕЛЕНИЕ КОЛЛИЗИЙ АППРОКСИМИРУЮЩИХ СФЕР И ПРЯМОУГОЛЬНЫХ ПАРАЛЛЕЛЕПИПЕДОВ В СИСТЕМАХ ТРЕХМЕРНОГО МОДЕЛИРОВАНИЯ

(Исследования выполнены при поддержке РФФИ, грант № 13-07-00708)

*А.М. Трушин, научный сотрудник, leha_trushin@mail.ru
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)*

В системах трехмерного моделирования виртуальные объекты могут сталкиваться друг с другом. Определение таких столкновений (коллизий) является неотъемлемой частью любого физического движка. Для физических движков важнейшую роль играет скорость их расчетов. Для поддержки режима реального времени расчеты одного кадра моделирования не должны превышать 40 мс для обеспечения визуализации не менее 25 кадров в секунду, поэтому к системе расчета динамики в целом и к определениям коллизий в частности предъявляется требование разработки быстрых и эффективных алгоритмов. Определение коллизий объектов произвольной формы – трудная задача, имеющая высокую вычислительную сложность, поэтому широко применяется метод определения коллизий при помощи аппроксимирующих контейнеров. В этом случае формы виртуальных объектов описываются (аппроксимируются) различными геометрическими примитивами и задача определения коллизий самих объектов сводится к определению коллизий их аппроксимирующих контейнеров. Широкое распространение получили такие примитивы, как прямоугольные параллелепипеды и сферы. Алгоритмы определения коллизий бывают априорными и апостериорными. Априорные алгоритмы предсказывают коллизии тел, а апостериорные определяют коллизии уже по факту пересечения самих объектов. При этом априорные алгоритмы в общем случае обладают значительно большей вычислительной сложностью ввиду большого объема входных данных. В связи с этим в физических движках, ориентированных на моделирование динамики в режиме реального времени, в основном используются апостериорные алгоритмы определения коллизий. Настоящая работа посвящена разработке быстрых и эффективных апостериорных алгоритмов определения коллизий сфер между собой и сфер с прямоугольными параллелепипедами.

Ключевые слова: *определение коллизий, аппроксимирующие контейнеры, моделирование динамики.*

К важнейшим задачам при моделировании динамики трехмерных виртуальных объектов (тел) относятся определение и обработка их столкновений (коллизий) [1]. Определение коллизий объектов произвольных форм является вычислительно сложной задачей [2], поэтому широкое распространение получил метод определения коллизий, при котором каждый динамический объект окружается (аппроксимируется) одним или несколькими геометрическими контейнерами (параллелепипедами, сферами и т.д.) [3]. Тогда моделирование столкновения двух тел заключается в определении параметров (точки, глубины и нормали) коллизии аппроксимирующих контейнеров двух объектов и в задании реакции тел на это столкновение.

Алгоритмы определения коллизий делятся на два вида: априорные и апостериорные [4]. Априорные алгоритмы предсказывают столкновения тел, апостериорные определяют коллизию объектов уже по факту их столкновения. В общем случае априорные алгоритмы обладают большей точностью вычисления информации о коллизии, но их существенным недостатком является большая вычислительная сложность, связанная с большим объемом входных данных (положение и ориентация объектов, а также их физические параметры: скорость, сила, моменты и т.д.). В связи с этим системы динамики, основанные на априорных определениях коллизий, работают в режиме реального времени (не менее 25 шагов моделирования в секунду) лишь для небольшого числа виртуальных объектов. Для апостериорных алгоритмов опреде-

ления коллизий входной объем данных сводится только к положению и ориентации объектов.

В настоящей работе рассматриваются методы апостериорного определения коллизий двух сфер и сферы с прямоугольным параллелепипедом (боксом). Коллизии боксов между собой рассмотрены в [5].

Понятие коллизии

В данной работе будут использоваться правосторонние системы координат. Для каждого аппроксимирующего контейнера задается матрица перехода из его локальной системы координат в мировую [6]. Первые три столбца этой матрицы задают ориентацию тела, а четвертый – положение начала локальной системы координат тела в мировой системе координат.

Обозначим радиус сферы через r , а начало ее локальной системы координат расположим в центре сферы.

Размеры бокса задаются величинами l (половина длины), w (половина ширины) и h (высота). Начало локальной системы координат расположим в центре нижней грани бокса так, что ось x направлена параллельно ребру бокса, задающему его длину, ось y – ширину, а ось z – высоту.

Под определением коллизии пары аппроксимирующих контейнеров понимается определение факта их пересечения и, в случае его наличия, вычисление точки коллизии, нормали расталкивания и глубины коллизии.

Точка, нормаль и глубина коллизии. В качестве первого объекта коллизии всегда будем рассматривать сферу, а вторым объектом будет либо сфера, либо бокс. Определим точку коллизии как точку поверхности второго объекта, наиболее близкую к центру первого. Под нормалью коллизии будем подразумевать единичный вектор, направленный вдоль прямой, проходящей через точку контакта и центр первого объекта, и указывающий направление расталкивания для первого объекта. Глубина коллизии – это расстояние, на которое точка коллизии проникла в первый объект вдоль нормали.

Обозначим через P точку коллизии. Пусть O_1 и O_2 – положение центров первого и второго объектов в мировой системе координат, а M_1 и M_2 – соответствующие матрицы перехода. Если вторым объектом является сфера, то ее центр задается четвертым столбцом матрицы M_2 , если же бокс, то к координате z четвертого столбца надо прибавить $h/2$ (так как начало системы координат бокса располагается в центре нижнего основания). Тогда нормаль коллизии определим вектором

$$\vec{N} = \frac{(\overrightarrow{O_2 O_1}, \overrightarrow{P O_1}) \overrightarrow{P O_1}}{|(\overrightarrow{O_2 O_1}, \overrightarrow{P O_1}) \overrightarrow{P O_1}|}, \quad (1)$$

а глубину коллизии величиной

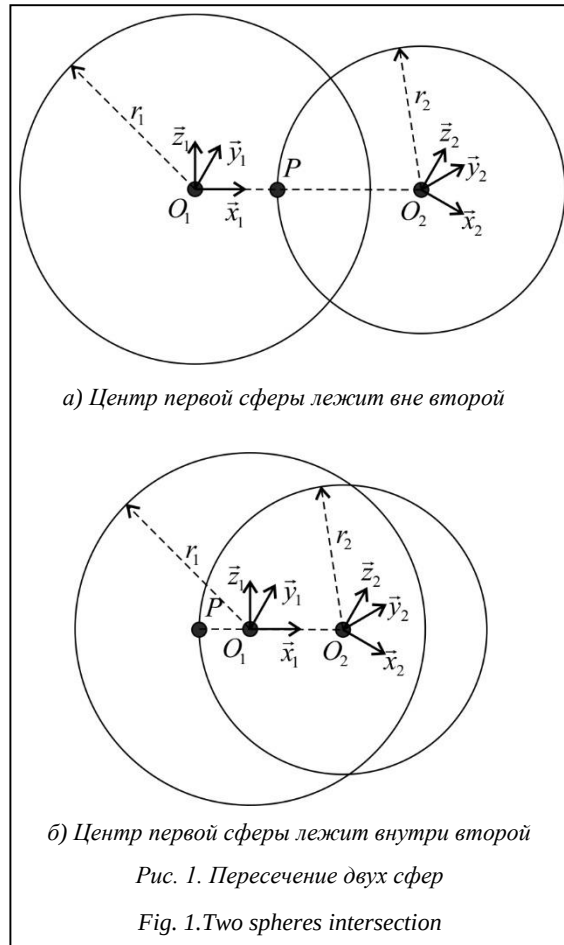
$$D = r_1 - (\vec{N}, \overrightarrow{P O_1}). \quad (2)$$

Определение коллизии двух сфер

Две сферы пересекаются тогда и только тогда, когда сумма их радиусов больше или равна расстоянию между их центрами (см. рис. 1), то есть $|O_1 - O_2| \leq r_1 + r_2$.

Если сферы пересекаются, то точка коллизии лежит на поверхности второй сферы на прямой, соединяющей их центры: $P = O_2 + r_2 \frac{\overrightarrow{O_2 O_1}}{|\overrightarrow{O_2 O_1}|}$.

Проверим совпадение направления нормали (1) с направлением вектора $\overrightarrow{O_2 O_1}$, который и будет вектором, задающим правильное направление для расталкивания первого объекта. Рассмотрим два случая: центр O_1 находится вне второй сферы (рис. 1а) и центр O_1 находится внутри второй сферы (рис. 1б). В первом случае векторы $\overrightarrow{P O_1}$ и $\overrightarrow{O_2 O_1}$ направлены в одну сторону, поэтому $(\overrightarrow{O_2 O_1}, \overrightarrow{P O_1}) > 0$ и, следовательно, вектор $(\overrightarrow{O_2 O_1}, \overrightarrow{P O_1}) \overrightarrow{P O_1}$ также совпадает по направлению с вектором $\overrightarrow{O_2 O_1}$, то есть направление нормали верное. Во втором случае вектор $\overrightarrow{P O_1}$ направлен противоположно вектору $\overrightarrow{O_2 O_1}$, поэтому $(\overrightarrow{O_2 O_1}, \overrightarrow{P O_1}) < 0$. Произведение $(\overrightarrow{O_2 O_1}, \overrightarrow{P O_1}) \overrightarrow{P O_1}$



даст вектор, совпадающий по направлению с вектором $\overrightarrow{O_2 O_1}$, то есть нормаль направлена верно.

Тогда из (2) получаем, что глубина коллизии в первом случае будет $D = r_1 - (\vec{N}, \overrightarrow{P O_1}) = r_1 - |\overrightarrow{P O_1}|$, а во втором – $D = r_1 - (\vec{N}, \overrightarrow{P O_1}) = r_1 + |\overrightarrow{P O_1}|$.

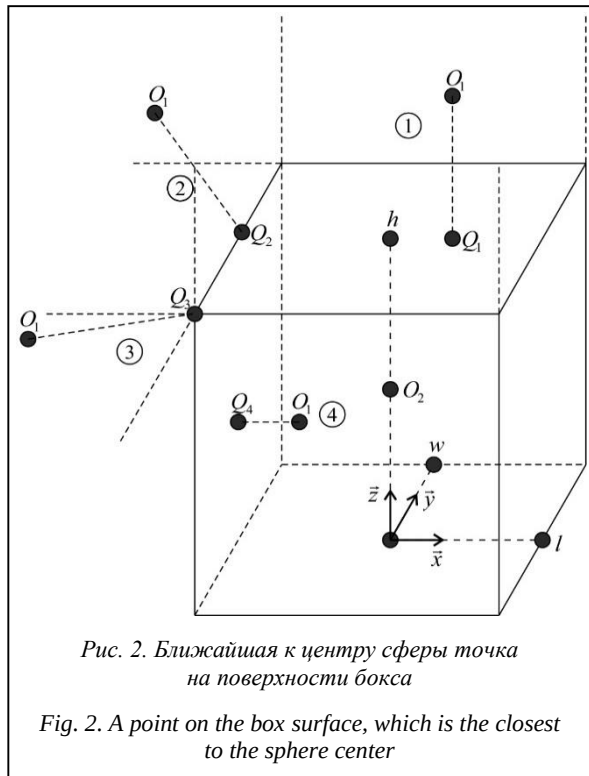
Определение коллизии сферы с прямоугольным параллелепипедом

Для определения коллизии сферы и бокса обозначим через Q точку на поверхности бокса, ближайшую к центру сферы. Рассмотрим всевозможные варианты расположения центра сферы относительно бокса (см. рис. 2).

1. Центр сферы расположен точно над одной из граней бокса. Тогда точкой Q является конец перпендикуляра, опущенного из центра сферы к этой грани (точка Q_1 на рисунке 2).

2. Центр сферы расположен по диагонали от одного из ребер бокса, то есть вне двух соседних граней, содержащих это ребро. Тогда точкой Q является конец перпендикуляра, опущенного из центра сферы к этому ребру бокса (точка Q_2 на рисунке 2).

3. Центр сферы расположен по диагонали от одной из вершин бокса, то есть вне трех соседних гра-



ней, содержащих эту вершину. Тогда точкой Q будет эта вершина бокса (точка Q_3 на рисунке 2).

4. Центр сферы лежит внутри бокса. Тогда точкой Q является конец перпендикуляра, опущенного из центра сферы к ближайшей грани бокса (точка Q_4 на рисунке 2).

Обозначим через $E1$ систему координат сферы, а через $E2$ систему координат бокса. Рассмотрим достаточно быстрый алгоритм, позволяющий определить положение центра сферы относительно бокса. Причем, если центр сферы лежит вне бокса, алгоритм сразу вычисляет точку Q . Суть алгоритма – перевод вектора $\overrightarrow{O_2O_1}$ из мировой системы координат в систему $E2$ и его покомпонентное сравнение с полуразмерами бокса. Обозначим совокупность координат вектора $\overrightarrow{O_2O_1}$ в системе координат $E2$ через $\vec{u}_{E2}$. Получим $\vec{u}_{E2} = M_2^{-1}(O_1 - O_2)$.

Тогда с использованием компонент x , y и z вектора $\vec{u}_{E2}$ алгоритм будет следующим.

Если $\vec{u}_{E2}.x < -l/2$, то

$$\vec{u}_{E2}.x = -l/2,$$

иначе, если $\vec{u}_{E2}.x > l/2$, то

$$\vec{u}_{E2}.x = l/2.$$

Если $\vec{u}_{E2}.y < -w/2$, то

$$\vec{u}_{E2}.y = -w/2,$$

иначе, если $\vec{u}_{E2}.y > w/2$, то

$$\vec{u}_{E2}.y = w/2.$$

Если $\vec{u}_{E2}.z < -h/2$, то

$$\vec{u}_{E2}.z = -h/2,$$

иначе, если $\vec{u}_{E2}.z > h/2$, то

$$\vec{u}_{E2}.z = h/2.$$

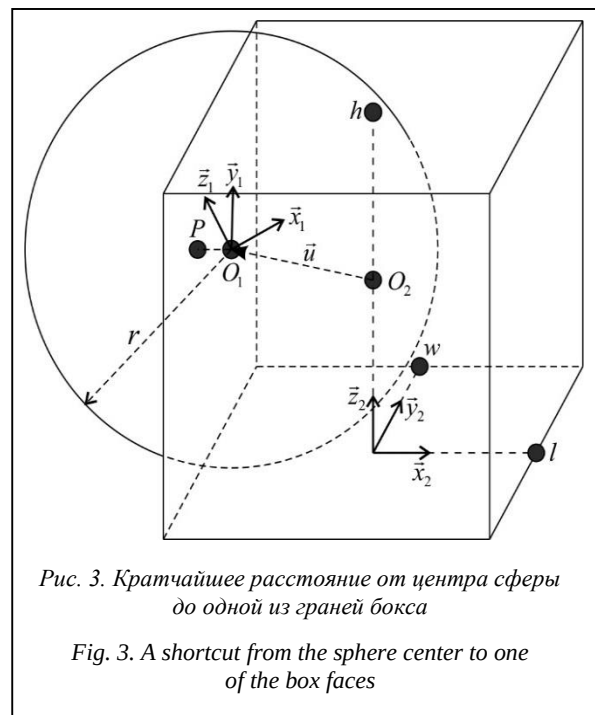
Если выполняется хотя бы одно из данных условий, значит, центр сферы лежит вне бокса. Тогда точка Q в системе координат $E2$ будет $Q_{E2} = O_{2E2} + \vec{u}_{E2}$, а в мировой $Q = M_2(O_{2E2} + \vec{u}_{E2}) = O_2 + M_2\vec{u}_{E2}$.

Если не выполнилось ни одно из условий, значит, центр сферы лежит внутри бокса. Рассмотрим отдельно оба случая.

Центр сферы лежит вне бокса. Сфера и бокс пересекаются тогда и только тогда, когда радиус сферы больше расстояния от ее центра O_1 до точки Q , то есть $|O_1 - Q| \leq r_1$.

Если сфера и бокс пересекаются, точка коллизии P будет совпадать с Q . Ясно, что соответствующие координаты векторов $\overrightarrow{O_2O_1}$ и $\overrightarrow{PO_1}$ в системе $E2$ имеют одинаковые знаки, поэтому $(\overrightarrow{O_2O_1}, \overrightarrow{PO_1}) > 0$. Тогда из (1) следует, что $\vec{N}$ совпадет по направлению с $\overrightarrow{PO_1}$ (то есть будет соответствовать направлению расталкивания сферы), а из (2) получаем $D = r_1 - (\vec{N}, \overrightarrow{PO_1}) = r_1 - |\overrightarrow{PO_1}|$.

Центр сферы лежит внутри бокса. В этом случае вектор $\vec{u}_{E2}$ будет соединять центр бокса с центром сферы (см. рис. 3).



Вычислим направление $\vec{d}$ и расстояние $\min$ от центра сферы до ближайшей грани бокса:

$$\min = l/2 - |\vec{u}_{E2}.x|, \vec{d} = (\vec{u}_{E2}.x \ 0 \ 0)^T.$$

Если $\min > w/2 - |\vec{u}_{E2}.y|$, то

$$\min = w/2 - |\vec{u}_{E2}.y|, \vec{d} = (0 \ \vec{u}_{E2}.y \ 0)^T.$$

Если $\min > h/2 - |\vec{u}_{E2} \cdot \vec{z}|$, то

$$\min = h/2 - |\vec{u}_{E2} \cdot \vec{z}|,$$

$$\vec{d} = (0 \ 0 \ \vec{u}_{E2} \cdot \vec{z})^T.$$

Тогда точка коллизии

$$P = O_1 + \min \cdot \frac{\vec{d}}{|\vec{d}|}.$$

Проверим, что направление нормали (1) будет противоположно направлению вектора $\vec{PO}_1$, то есть будет правильным для расталкивания первого объекта. Ясно, что косинус угла между векторами $\vec{O_2O_1}$ и $\vec{PO}_1$ отрицательный, то есть $(\vec{O_2O_1}, \vec{PO}_1) < 0$. Следовательно, направление нормали будет противоположно направлению $\vec{PO}_1$. Тогда из (2) получаем, что в этом случае глубина коллизии $D = r_1 - (\vec{N}, \vec{PO}_1) = r_1 + |\vec{PO}_1|$.

В заключение отметим, что рассмотренные алгоритмы определения коллизий двух сфер между собой и сферы с прямоугольным параллелепипедом позволяют вычислять информацию о коллизиях с небольшой вычислительной сложностью. Вместе с обработкой коллизий, описанной в [7], это позволяет моделировать динамику в режиме реального времени большого (вплоть до нескольких тысяч) количества виртуальных объектов. Данные алгоритмы были реализованы в рамках динамической библиотеки, входящей в состав существующей системы трехмерного моделирования виртуальных объектов. За счет наличия в этой системе модуля визуализации [8, 9] реализованные алгоритмы были проверены на наборе тестовых сцен, имитирующих различные ситуации при столкновении виртуальных объектов. Особое внимание было уделено сложной в плане моделирования динамики ситуации покоя множества тел друг на друге (см. рис. 4). Сложность такой ситуации при моделировании динамики в том, что на тела постоянно действует сила тяжести. Таким образом, тела постоянно стремятся проникать друг в друга, а подсистема определения и обработки коллизий этому противодействует. До тех пор, пока в этой подсистеме получаемые расчеты по обработке коллизий не полностью удовлетворяют условию покоя всех тел одновременно, объекты совершают небольшие движения вверх-вниз (подрагивают).



Рис. 4. Множество тел друг на друге в состоянии покоя

Fig. 4. Many motionless objects one on another

Система динамики в примере покоя объектов друг на друге показала убедительные результаты времени расчетов. На просчет одного шага моделирования при наличии незначительного дрожания объектов уходит не более 3 мс. В течение небольшого количества шагов моделирования объекты перестают дрожать и переходят в статическое положение, при котором определение и обработка их коллизий прекращаются и на просчет одного шага моделирования уходит менее 1 мс.

Литература

1. Moore M., Wilhelms J. Collision Detection and Response for Computer Animation. Computer Graphics, 1998, vol. 22, no. 4, pp. 289–298; URL: <http://www.cs.princeton.edu/courses/archive/spring01/cs598b/papers/moore88.pdf> (дата обращения: 10.07.2015).

2. Ming C. Lin, Gottschalk S. Collision detection between geometric models: a survey. *Proc. of IMA conference on mathematics of surfaces*. 1998, vol. 1, pp. 602–608; URL: <https://users.soe.ucsc.edu/~pang/161/w06/notes/cms98.pdf> (дата обращения: 10.07.2015).
3. Bounding volume. URL: https://en.wikipedia.org/wiki/Bounding_volume (дата обращения: 10.07.2015).
4. Ericson C. *Real-Time Collision Detection*. CRC Press, 2004, 594 p.; URL: <https://books.google.ru/books?id=WGpL6Sk9qNAC> (дата обращения: 10.07.2015).
5. Михайлюк М.В., Трушин А.М. Расчет коллизий прямоугольных параллелепипедов в задачах динамики // Труды НИИСи РАН. Т. 2. № 2. Обработка изображений, моделирование и визуализация: теоретические и прикладные аспекты. 2012. С. 51–59.
6. Transformation matrix. URL: https://en.wikipedia.org/wiki/Transformation_matrix (дата обращения: 10.07.2015).
7. Трушин А.М. Обработка коллизий виртуальных объектов с помощью метода последовательных импульсов // Труды НИИСи РАН. Т. 4. № 2. Математическое и компьютерное моделирование сложных систем: теоретические и прикладные аспекты. 2014. С. 95–105.
8. Михайлюк М.В., Торгашев М.А. Система «GLVIEW» визуализации для моделирующих комплексов и систем виртуальной реальности // Вестн. РАЕН. 2011. № 2. С. 20–28.
9. Михайлюк М.В., Торгашев М.А. Визуализация динамики объектов управления в реальном времени // Научная визуализация. 2014. Т. 6. № 5. С. 69–80; URL: <http://sv-journal.org/2014-5/index2139.html?lang=ru> (дата обращения: 10.07.2015).

DOI: 10.15827/0236-235X.112.105-109

Received 27.07.15

COLLISION DETECTION FOR BOUNDING SPHERES AND RECTANGULAR PARALLELEPIPEDS IN 3D MODELING SYSTEMS

(The research has been supported by RFBR, grant no. 15-07-04544)

Trushin A.M., Research Associate, leha_trushin@mail.ru;

(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation)

Abstract. Virtual objects in 3D modeling systems may collide with each other. Collision detection is an integral part of any physical engine. The speed of calculation is crucial for physical engines. In real-time mode one simulation frame calculations should not exceed 40 ms to visualize at least 25 frames per second. Therefore, there is a need in development of fast and efficient algorithms for the dynamics calculation system and for collision detection in particular. Collision detection of complex shape's objects is a difficult task, which has a high computational complexity. Therefore, a method using bounding volumes is widely used. In this case, virtual objects forms are described by as different geometric primitives, and the problem of objects' collision detection is reduced to the collision detection of their bounding volumes. Such primitives as rectangular parallelepipeds (boxes) and spheres became widespread. Collision detection algorithms may be a priori and a posteriori. A priori algorithms predict collisions of bodies, and a posteriori algorithms detect collisions after actual intersections of the objects. In general, a priori algorithms have much higher computational complexity due to the greater amount of input data. In this regard, physics engines oriented on real-time dynamics modeling basically use a posteriori collision detection algorithms. This work is devoted to the development of fast and efficient algorithms for a posteriori sphere-sphere and sphere-box collision detection.

Keywords: collision detection, bounding volumes, dynamics modeling.

References

1. Moore M., Wilhelms J. Collision Detection and Response for Computer Animation. *Computer Graphics*. 1998, vol. 22, no. 4, pp. 289–298. Available at: <http://www.cs.princeton.edu/courses/archive/spring01/cs598b/papers/moore88.pdf> (accessed July 10, 2015).
2. Ming C. Lin, Gottschalk S. Collision detection between geometric models: a survey. *Proc. of IMA Conf. on Mathematics of Surfaces*. 1998, vol. 1, pp. 602–608. Available at: <https://users.soe.ucsc.edu/~pang/161/w06/notes/cms98.pdf> (accessed July 10, 2015).
3. Bounding volume. *Wikipedia. The Free Encyclopedia*. Available at: https://en.wikipedia.org/wiki/Bounding_volume (accessed July 10, 2015).
4. Ericson C. *Real-Time Collision Detection*. CRC Press, 2004, 594 p. Available at: <https://books.google.ru/books?id=WGpL6Sk9qNAC> (accessed July 10, 2015).
5. Mikhaylyuk M.V., Trushin A.M. Computation of rectangular parallelepipeds collision detection in dynamics. *Trudy NIISI RAN* [Pros. of SRISA RAS]. Moscow, 2012, vol. 2, no. 2, pp. 51–59 (in Russ).
6. Transformation matrix. *Wikipedia. The Free Encyclopedia*. Available at: https://en.wikipedia.org/wiki/Transformation_matrix (accessed July 10, 2015).
7. Trushin A.M. Collision response of virtual objects using the method of sequential impulses. *Trudy NIISI RAN* [Pros. of SRISA RAS]. Moscow, 2014, vol. 4, no. 2, pp. 95–105 (in Russ).
8. Mikhaylyuk M.V., Torgashev M.A. “GLView” visualization system for simulation complexes and virtual reality systems. *Vestnik RAEN* [Bulletin of RANS]. 2011 no. 2, pp. 20–28 (in Russ).
9. Mikhaylyuk M.V., Torgashev M.A. Real-time visualization of controlled objects' dynamics. *Nauchnaya vizualizatsiya* [Scientific Visualization]. 2014, vol. 6, no. 5, pp. 69–80 (in Russ). Available at: <http://sv-journal.org/2014-5/index2139.html?lang=ru> (accessed July 10, 2015).

УДК 004.3

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.110-115

МАРШРУТ ПРОЕКТИРОВАНИЯ САМОСИНХРОННЫХ КОНВЕЙЕРНЫХ СХЕМ С ИСПОЛЬЗОВАНИЕМ ВОЗМОЖНОСТЕЙ САПР

(Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 14-29-09238 офи_м)

А.О. Власов, зав. сектором, vlaalex@cs.niisi.ras.ru;

А.В. Сурков, старший научный сотрудник, surkov@cs.niisi.ras.ru
(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)

Самосинхронная логика не использует сигнал тактирования и работает при любых вариациях параметров внешней среды, достаточных для переключения транзисторов. Но у самосинхронной логики есть ряд недостатков, из-за которых она не применяется наравне с традиционными синхронными схемами. Самосинхронные схемы проектируются двумя способами: на основе графовой спецификации и с использованием перекрестного преобразования. Первый способ дает наиболее быстрые и компактные схемы, но ввиду сложности работы с графами размер схемы ограничен несколькими десятками элементов. Метод перекрестного преобразования удобен при проектировании больших операционных блоков, однако в работе использует двухфазную дисциплину передачи данных и парафазное кодирование проводов, вследствие чего результирующая схема получает удвоенное число логических элементов. Как следствие, самосинхронные операционные блоки имеют как минимум вдвое меньшую скорость и в разы большую площадь схемы. Тем не менее, самосинхронная логика может использоваться там, где применение традиционных синхронных схем невозможно – прежде всего в расширенных диапазонах температур и вариации напряжения питания. Главным недостатком самосинхронных схем является отсутствие автоматизированного маршрута проектирования (САПР), в результате чего самосинхронные схемы, как правило, проектируются вручную. В целях сокращения времени разработки в НИИСИ РАН выработан маршрут проектирования самосинхронных конвейерных схем с использованием созданных ранее синхронных прототипов, включающий в себя синтез, разработку топологии и функциональное моделирование.

Ключевые слова: асинхронные, самосинхронные, конвейерные схемы, САПР, Design Compiler, Encounter.

Первая эффективная методика построения асинхронных конвейерных схем была предложена А. Сазерлендом в 1989 году [1]. Он представил конструкцию асинхронного микроконвейера, с помощью которого можно строить асинхронные схемы достаточно высокой сложности. Особенность его разработки в разделении схемы на управляющую и операционную части. При этом направления переключений в схеме управления и движения данных в операционной части были противоположно направленными (counter flow). В отличие от синхронных схем, где тактовый импульс приходит практически одновременно на все управляемые устройства (триггеры), осуществляя таким образом их параллельную синхронизацию, в микроконвейерах Сазерленда источник возбуждения всегда находится в начале конвейера, а информация распространяется волнами, поэтому синхронизация такого типа получила название волновой. При желании конвейер можно замкнуть – в этом случае волновое движение данных происходит циклически, без остановки. На основе микроконвейеров Сазерленда было сделано несколько разработок микропроцессоров, некоторые из которых выпускались серийно.

Методика построения микроконвейеров состояла в переделке синхронной схемы с помощью так называемой операции десинхронизации, когда дерево тактового сигнала в микросхеме заменялось локальными взаимодействиями типа «запрос-ответ» (handshake). Если рассмотреть одну стадию конвейера, состоящую из элементов ло-

гики и элементов памяти на ее выходах, то взаимодействие запрос-ответ строилось следующим образом: запрос формировался одновременно с подачей новых данных на вход логики, а ответ возвращался по защелкивании этих данных в элементах памяти после прохождения через логику. Чтобы избежать явления гонок, запрос необходимо было задержать на время, не меньшее, чем затрачиваемое на прохождение сигнала через логику. Для этого Сазерленд использовал моделирующие линии задержки (Bundled Delay), а для координации взаимодействия запрос-ответ соседних стадий конвейера использовался синхронизатор в виде С-элемента Маллера. Таким образом, основная сложность проектирования заключалась в расчете линии задержки для каждой из стадий микроконвейера, на базе которых проектировалось устройство. Следствием такого подхода был весьма ограниченный диапазон температур и напряжений, поскольку при выходе из диапазона значение задержки распространения сигнала в логике могло превысить моделирующую задержку и схема сбивалась. Вследствие этого схемы на базе микроконвейеров Сазерленда зависят от разброса задержек в элементах и проводах в той же степени, что и синхронные прототипы, из которых они переделывались. В НИИСИ РАН в основном исследуются самосинхронные схемы, которые выгодно отличаются от асинхронной архитектуры микроконвейеров Сазерленда своей способностью работать в расширенных диапазонах температур и питающих напряжений, поскольку

следствием самосинхронности является независимость от задержек в элементах, а при соблюдении правила эквихронной зоны еще и независимость от задержек в проводах.

Идея разделения асинхронной схемы на подсхему управления и операционную часть получила развитие в работе [2]. Был предложен новый подход к построению систем с глобальной асинхронной синхронизацией. Система рассматривается как совокупность взаимодействующих блоков, для которых организовано локальное управление – синхронизация, основанная на взаимодействии запрос-ответ. Подсистема управления (синхронизации) получила название синхростратум и по своим свойствам является самосинхронной схемой. Операционная часть, состоящая из управляемых взаимодействующих блоков, названа процессорным стратумом. При этом архитектура блоков необязательно должна быть самосинхронной; допускается схемотехника и на основе микроконвейеров Сазерленда (то есть с моделирующими линиями задержек), также это могут быть и синхронные блоки. Единственное условие, предъявляемое к управляемым блокам, – формирование сигнала ответа по завершении всех переходных процессов в блоке. Системы с синхростратумом, управляющим синхронными блоками, известны как GALS (Globally Asynchronous Locally Synchronous), а более общий случай с произвольной архитектурой взаимодействующих блоков называется GALA (Globally Asynchronous Locally Arbitrary). GALA-системы наиболее универсальны, поскольку осуществляют взаимодействие блоков с различной архитектурой. Существует и другая системная архитектура, когда в синхронное окружение вставляется, к примеру, самосинхронный операционный блок. Примером таких систем служит совместная разработка НИИСИ РАН и ИПИ РАН, где в синхронный микропроцессор КОМДИВ-64 встроены квазисамосинхронный блок деления и извлечения корня двойной точности [3].

Синхростратум строится по модульному принципу, где один модуль осуществляет синхронизацию одного блока и связан с соседними модулями согласно графу соединений по линиям данных между управляемыми блоками. При этом в качестве управляемого блока может выступать не только большой операционный блок со сложной внутренней структурой, но и всего лишь одна стадия конвейера, состоящая из логики и элементов памяти на выходе. В общем случае в качестве управляемого блока может выступать автомат Мура.

Таким образом, синхростратум может делиться на верхний уровень взаимодействий на уровне системы и управляемых блоков и более низкие по иерархии уровни, осуществляющие взаимодействие внутри блоков, подблоков и т.д.

Особенностью маршрута проектирования больших самосинхронных операционных блоков является введение парафазного кодирования сигналов и двухфазной дисциплины передачи данных. Парафазное кодирование требуется для придания свойства монотонности логической функции операционного блока, а двухфазная дисциплина передачи позволяет это свойство монотонности использовать. При парафазном кодировании значения сигнала 10 и 01 являются информационными и относятся к фазе данных, а значения 00 и 11 относятся к фазе спейсера и являются служебными (спейсерными).

Синхростратумы делятся по способу синхронизации на два типа: синхростратумы с волновой синхронизацией и с параллельной синхронизацией. Для конвейерных схем целесообразно использовать волновую синхронизацию, поскольку информация в конвейере распространяется однонаправленно от входа к выходу по принципу волны. Особенность асинхронных конвейерных схем с волновой синхронизацией состоит в том, что конвейер может иметь различную степень наполненности – от пустого конвейера до плотно наполненного. При пустом конвейере первые загруженные данные имеют самую высокую скорость прохождения через конвейер. При частично наполненном конвейере наибольшая скорость достигается при работе соседних стадий конвейера в противофазе. Максимальное наполнение конвейера может быть достигнуто остановкой выгрузки данных с его выходов. При использовании специальных схемотехнических приемов [4] в случае останова конвейера максимальное наполнение может достигать 100 %, это означает, что в каждой стадии конвейера хранится уникальная информация; для сравнения: в схеме Сазерленда максимальное наполнение достигает всего 50 %, что означает уникальность хранимой информации только в каждой второй стадии конвейера.

Маршрут проектирования самосинхронных конвейерных схем

В НИИСИ РАН разработан маршрут автоматизированного проектирования самосинхронных конвейерных схем. Как показала работа коллег из ИПИ РАН, наилучшие результаты получаются при проектировании вручную. Недостатком такого подхода являются сроки – на разработку одного конвейерного блока уходит около года. В этой связи необходимо было разработать автоматизированный маршрут проектирования, использующий уже существующие синхронные наработки и существенно сокращающий время проектирования.

Для физического проектирования цифровых устройств в НИИСИ используются такие популярные САПР комплекса, как Design Compiler

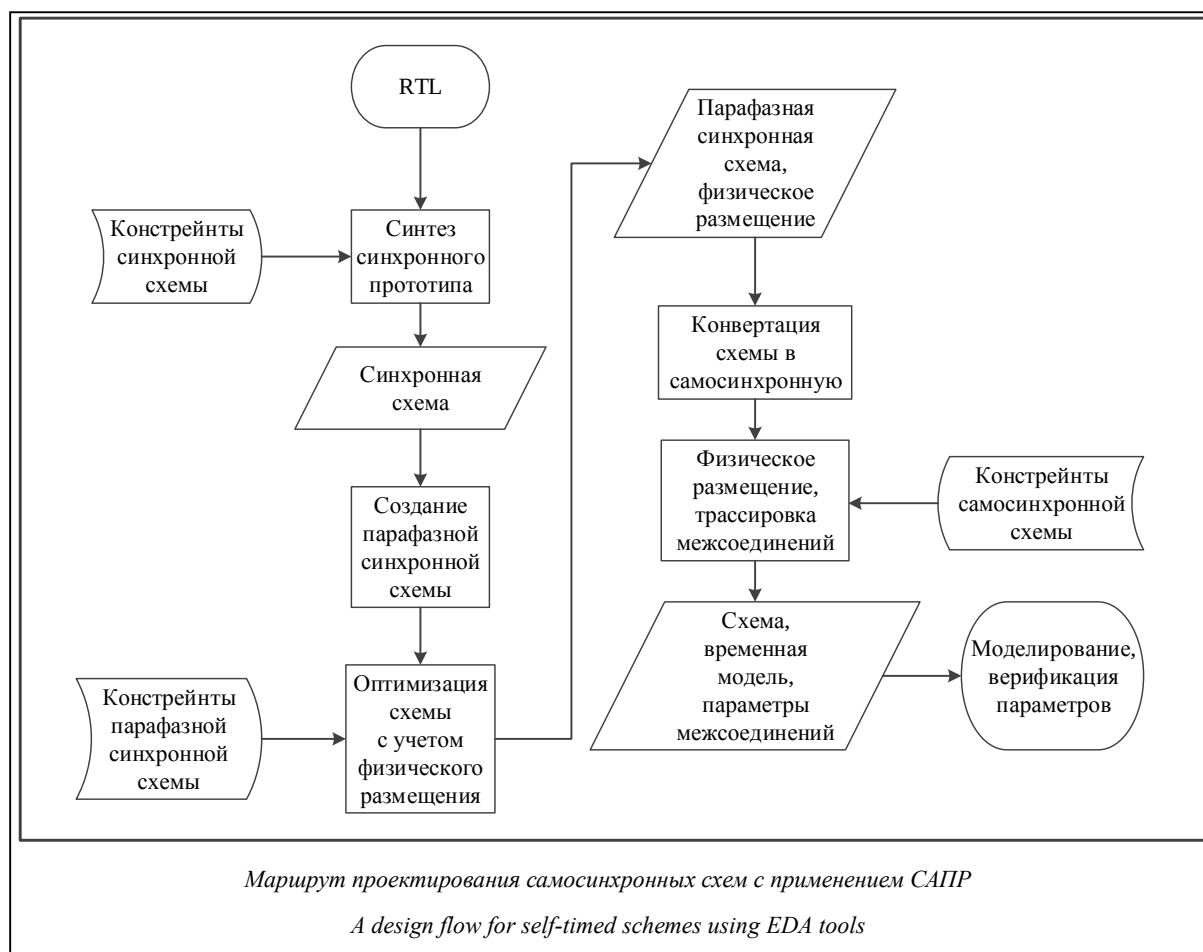
(Synopsys) и Encounter (Cadence). В соответствии с используемыми САПР программами стандартный маршрут физической имплементации синхронных устройств состоит из двух этапов:

- схемотехнический синтез RTL в базе используемых библиотечных элементов (Design Compiler);
- физическое размещение, построение системы синхронизации, трассировка межсоединений (Encounter).

Фактически в процессе проектирования осуществляется преобразование логического описания RTL в топологию для производства (GDS). Существующие в САПР программы ориентированы на проектирование синхронных устройств. Для физической реализации самосинхронных схем непосредственное их применение невозможно. Однако выборочное использование их функционала дает возможность построить маршрут физического автоматизированного проектирования самосинхронных схем. Этапы такого маршрута показаны на рисунке.

Целесообразно использовать существующие синхронные наработки, поскольку экономится время на разработку логической модели, а также существенно сокращаются сроки тестирования и исправления ошибок. Идея о возможности пере-

делки любой синхронной схемы в самосинхронную впервые изложена в [2]. Синтезированную схему синхронного прототипа можно переделать в самосинхронную схему, используя шаблоны. Для переделки блоков логики удобно использовать перекрестное преобразование [5], которое в процессе парафазного представления сигналов и логики строит прямой и инверсный каналы передачи данных. Также в процессе перекрестного преобразования становятся монотонными функции всех содержащихся в прямом и инверсном каналах пар логических элементов, что достигается коррекцией входных наборов на входах логических элементов с помощью дополнительных инверторов и перекрестных связей между каналами. А поскольку результирующая функция блока логики является суперпозицией функций составляющих ее элементов, результирующая функция также становится монотонной. В процессе преобразования синхронной схемы D-триггеры заменяются самосинхронными защелками, а дерево тактового сигнала – синхростратумом, осуществляющим волновую синхронизацию, и схемой индикации окончания переходных процессов в схеме. Все операции по преобразованию синхронного прототипа в самосинхронную схему осуществляются в автоматическом режиме с помо-



щью скрипта. В самосинхронной схеме используются асинхронные элементы, такие как C-элемент Миллера, которых нет в используемых библиотеках стандартных элементов. Эти элементы были созданы в САПР Virtuoso и характеризованы в САПР SiliconSmart. Таким образом, полученная в результате работы скрипта схема представляет собой нетлист в базисе как стандартных, так и асинхронных элементов и может использоваться при проектировании топологии.

Учитывая вышесказанное, отметим, что первым шагом маршрута является схемотехнический синтез RTL-описания с целью получения исходной синхронной схемы. Для этого используются Design Compiler и стандартные шаги, характерные для проектирования синхронных схем. Контроль временных параметров задается файлом с описанными на языке tcl временными ограничениями – констрейнтами.

Полученная схема с помощью программы, написанной на языке Perl, преобразуется в парафазную синхронную схему. Для нее создаются новые констрейнты, отражающие внесенные изменения. После модификации схемы требуется ее оптимизация с учетом влияния параметров межсоединений. Учитывая, что парафазный вариант, как уже говорилось, – лишь подготовка к конвертации в самосинхронный вариант, решено произвести физическое размещение данной схемы с последующей оптимизацией. В связи с тем, что для программы конвертации важно сохранение элементной структуры парафазной схемы при оптимизациях, было внесено ограничение на изменения схемы – запретить пересинтезирование участков цепи. Результатом этого этапа стали оптимизированная с учетом параметров межсоединений парафазная схема и файл с информацией о размещении ее элементов (DEF-файл).

Следующие шаги маршрута – конвертация программой, написанной на языке Perl, парафазной схемы в самосинхронную и последующая ее физическая реализация с использованием информации о размещении элементов парафазной схемы. Программа конвертации изменяет небольшое количество элементов парафазной схемы и добавляет элементы системы самосинхронизации. При размещении большая часть информации о размещении элементов парафазной схемы остается актуальной, что ускоряет работу САПР и делает процесс проектирования последовательным. Для реализации статического временного анализа схемы были предприняты следующие меры: сигнал сброса Reset объявляется синхросигналом, а для элементов, которые являются его потребителями, были сделаны два библиотечных описания: синхронное и асинхронное. Синхронное описание представляет эти элементы триггерами и позволяет САПР реализовать контроль временных параметров цепей схемы при проектировании.

Асинхронное описание используется лишь при выписке временной модели (SDF-файла) устройства. Таким образом, к оптимизированной парафазной схеме и DEF-файлу добавляются констрейнты для самосинхронной схемы и дополнительный библиотечный файл, отражающие вышесказанное. Данный набор позволяет осуществить в САПР Encounter три этапа топологического проектирования самосинхронной схемы:

- размещение; к уже размещенным ячейкам парафазной схемы добавляются элементы самосинхронной версии, и осуществляется оптимизация размещения всех элементов;

- построение системы синхронизации; строятся цепи распространения сигнала Reset, сама реальная система самосинхронизации при этом уже построена программой конвертации и оптимизирована во время этапа размещения;

- трассировка межсоединений; реализуется реальное (а не оценочное, как на предыдущих этапах) трассирование межсоединений с учетом конструкторских норм и более точных параметров.

Полученные после физической реализации три файла – схема, временная модель и экстракция параметров межсоединений – позволяют моделировать устройство с целью проверки работоспособности схемы и получения ее реальных временных параметров. Данные операции составляют финальный этап маршрута физического автоматизированного проектирования самосинхронных схем.

Пример физического проектирования самосинхронного блока умножения с накоплением

Разработанный маршрут проверен на задаче физической реализации блока умножения с накоплением двойной точности (FMA) в виде 4-стадийного самосинхронного конвейера с волновой синхронизацией. В качестве синхронного прототипа взят блок FMA, являющийся частью FPU процессора КОМДИВ-64 разработки НИИСИ РАН [6]. Проектирование производилось для техпроцесса фабрики TSMC 65nm. Финальные параметры разных реализаций показаны в таблице.

Параметры синхронной и самосинхронной реализации схемы

The parameters of synchronous and self-timed scheme implementation

Тип реализации блока	Период, нс	Площадь ячеек, мм ²	Статическая мощность, мВт
Синхронный прототип	0,89	0,19	3,2
Самосинхронная схема	2,6	0,8	12,1

Скорость работы самосинхронной конвейерной схемы удобно оценивать по длительности периода смены фаз работающего конвейера. Период смены фаз можно измерить двумя способами: моделированием и с помощью статического временного анализа; при этом наибольшая пропускная способность самосинхронного конвейера с волновой синхронизацией соответствует наполненности на одну четверть. В моделировании в установленном режиме все стадии конвейера работают со скоростью, равной скорости самой медленной из стадий. Таким образом, период смены фаз всей схемы можно принять равным периоду смены фаз в самой медленной из стадий. Аналогичный результат получается и при проведении статического временного анализа, поскольку результатом является длительность самого медленного пути распространения сигнала внутри стадии конвейера. Поскольку распространение информации внутри конвейера происходит волнами, соответствующими отсчетам логического времени [7], можно считать корректным сравнение скоростных показателей работы (период смены фаз с периодом тактового сигнала) самосинхронной схемы и ее синхронного прототипа. Согласно таблице, скорость самосинхронной схемы получается в 3 раза ниже, чем у синхронного аналога. Это объясняется двухфазной дисциплиной передачи данных, что автоматически удваивает время распространения сигнала через схему; влияет на скорость и схема сжатия индикации, задержка которой растет вместе с увеличением числа разрядов в стадии конвейера.

Суммарная площадь элементов самосинхронной схемы получилась примерно в 4 раза больше площади синхронного прототипа. Разница площадей также объясняется схемой сжатия индикации, но наибольшее влияние оказывает парафазное кодирование сигналов, требующее удвоенного числа элементов логики. Увеличение площади способствует пропорциональному увеличению статической мощности токов утечки в 4 раза. Рассчитать динамическую мощность потребления не удалось, поскольку стандартный способ, заключающийся в использовании результатов моделирования, не подходит ввиду отсутствия активности на сигнале сброса Reset, который используется в САПР в качестве псевдосинхросигнала. То же касается и второго способа измерения динамического потребления, заключающегося в измерении переключательной активности элементов во время тестов. Таким образом, оценка динамического потребления самосинхронных схем в САПР остается предметом дальнейших исследований.

На основании изложенного отметим следующее. Авторами описана разработка маршрута проектирования самосинхронных конвейерных схем с использованием коммерческих САПР, включающего в себя моделирование, логический синтез и

разработку топологии. Маршрут проектирования рассчитан на использование предварительно разработанных синхронных блоков, поэтому может использовать ранее разработанные блоки, экономия, таким образом, время на разработку, отладку и тестирование. Полученные самосинхронные блоки представляют собой полные функциональные аналоги своих синхронных прототипов с учетом таких особенностей, как парафазное кодирование проводов и двухфазная дисциплина передачи данных. На примере блока FMA проведено сравнение результатов преобразования синхронного прототипа в самосинхронный конвейер волновой синхронизацией. Скорость работы самосинхронного блока оказалась в 3 раза ниже, а площадь в 4 раза больше, чем у синхронного прототипа. Полученные результаты коррелируют с прогнозами предыдущих исследований [5].

Предметом дальнейших исследований являются динамическая мощность потребления самосинхронных конвейерных схем, а также возможность автоматического проектирования самосинхронных схем с неполной индикацией (так называемых квазисамосинхронных схем), которые следует рассматривать как некий оптимум между производительностью и скоростью, ценой значительного ограничения свойств самодиагностики и выявления неисправностей. Отдельного рассмотрения требуют и самосинхронные схемы с глобально параллельной синхронизацией, поскольку они лучше подходят для задачи построения автоматов, а не только конвейерных схем.

Литература

1. Sutherland I.E. Micropipelines. Communication of ACM, June 1989, vol. 32, no. 6, pp. 720–738.
2. Varshavsky V.I., Marakhovsky V.B., Chu T.A. Asynchronous Timing of Arrays with Synchronous Prototype. Proc. 2nd Intern. Conf. on Massively Parallel Computing Systems (MPCS'96), 1996, pp. 47–54.
3. Бобков С.Г., Горбунов М.С., Дьяченко Ю.Г., Рождественский Ю.В., Степченков Ю.А., Сурков А.В. Использование самосинхронной логики для снижения потребляемой мощности и повышения надежности микропроцессоров // Проблемы разработки перспективных микро- и наноэлектронных систем: VI Всеросс. науч.-технич. конф. (МЭС-2014): сб. науч. тр. М.: Изд-во ИППМ РАН, 2014. № 1. С. 43–48.
4. Бобков С.Г., Мараховский В.Б., Сурков А.В. Модуль синхростратума, координирующего взаимодействие асинхронных автоматов Мура при волновой обработке данных. Заявка на изобр. № 2015112623, 07.04.2015.
5. Сурков А.В. Использование Synopsys Design Compiler для синтеза самосинхронных схем // Программные продукты и системы. 2014. № 4. С. 24–30.
6. Степченков Ю.А., Рождественский Ю.В., Дьяченко Ю.Г., Морозов Н.В., Степченков Д.Ю., Сурков А.В. Самосинхронное устройство умножения-сложения гигафлопсного класса: варианты реализации // Проблемы разработки перспективных микро- и наноэлектронных систем: VI Всеросс. науч.-технич. конф. (МЭС-2014): сб. науч. тр. М.: Изд-во ИППМ РАН, 2014. Т. 4. С. 57–60.
7. Мараховский В.Б., Мелехин В.Ф. Проектирование средств синхронизации блоков глобально асинхронных систем с произвольной локальной синхронизацией // Информационно-управляющие системы. 2010. № 1. С. 29–38.

DOI: 10.15827/0236-235X.112.110-115

Received 27.07.15

AUTOMATED DESIGN FLOW OF SELF-TIMED PIPELINES USING EDA TOOLS*(The reported study was partially supported by RFBR, research projects № 14-29-09238)***Vlasov A.O.**, Head of Sector, vlaalex@cs.niisi.ras.ru;**Surkov A.V.**, Senior Researcher, surkov@cs.niisi.ras.ru*(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation)*

Abstract. The self-timed logic doesn't use the clock signal and operates in a whole range of environmental parameters, where transistors still can switch. But there are some weak points due to which the self-timed logic is not used along with the traditional synchronous logic. The design of self-timed schemes has two major approaches: signal transition graph (STG) design and the so-called burst-mode technique. The first approach provides the fastest and the smallest schemes, but due to high complexity of working with graphs the scheme size is limited by a number of elements. The burst-mode technique is the best to design big logical blocks, but requires the four-phase dual rail protocol, which leads to logic doubling in the resulting scheme. As a result, the burst-mode logical blocks operate at a half speed and occupy much higher area. Nevertheless, the self-timed logic may still be used in applications, which require extended temperature range and/or supply voltage variation, where the synchronous logic is mostly unavailable. The major disadvantage of all self-timed techniques is the absence of the automated design flow, so self-timed schemes are mostly implemented manually. To reduce the design time we have developed the design flow based on previously created synchronous prototypes. It includes the synthesis, layout and functional verification via modeling.

Keywords: asynchronous, self-timed, pipelines, EDA, Design Compiler, Encounter.

References

1. Sutherland I.E. Micropipelines. *Communication of ACM*. 1989, vol. 32, no. 6, pp. 720–738.
2. Varshavsky V.I., Marakhovsky V.B., Chu T.A. Asynchronous Timing of Arrays with Synchronous Prototype. *Proc. of the 2nd Int. Conf. on Massively Parallel Computing Systems (MPCS'96)*. 1996, pp. 47–54.
3. Bobkov S.G., Gorbunov M.S., Dyachenko Yu.G., Rozhdestvensky Yu.V., Stepchenkov Yu.A., Surkov A.V. Self-Timed Circuits for Low Power and Highly Reliable Microprocessors. *Sb. nauch. tr. Vseross. nauch.-tekhnich. conf. "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem" (MES-2014)* [Proc. of the 4th All-Russian Science and Technical Conf. "Development Issues of Advanced Micro and Nanoelectronic Systems"]. 2014, no. 1, pp. 43–48 (in Russ.).
4. Bobkov S.G., Marakhovsky V.B., Surkov A.V. *Modul sinkhrostratuma, koordiniruyushchego vzaimodeystvie asinkhronnykh avtomatov Mura pri volnovoy obrabotke dannykh* [A Module of a Synchron-stratum for Wave Data Processing]. Submitted to Russian Patent Service 07.04.2015, registration no. 2015112623.
5. Surkov A.V. Synthesis of burst-mode asynchronous schemes using Synopsys Design Compiler. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4, pp. 24–30 (in Russ.).
6. Stepchenkov Yu.A., Rogdestvensky Yu.V., Dyachenko Yu.G., Morozov N.V., Stepchenkov D.Yu., Surkov A.V. Speed-Independent Fused Multiply-Add Unit of Gigaflows Rating: Implementation Variants. *Sb. nauch. tr. Vseross. nauch.-tekhnich. conf. "Problemy razrabotki perspektivnykh mikro- i nanoelektronnykh sistem" (MES-2014)* [Proc. of the 4th All-Russian Science and Technical Conf. "Development Issues of Advanced Micro and Nanoelectronic Systems"]. 2014, no. 4, pp. 57–60 (in Russ.).
7. Marakhovsky V.B., Melekhin V.F. Design of Module Synchronization Facilities in Globally Asynchronous Multiprocessor Systems with Arbitrary Local Synchronization. St.-Petersburg. *Informatsionno-upravlyayushchie sistemy* [Information and Control Systems]. 2010, vol. 1, pp. 29–38 (in Russ.).

Поздравляем с днем рождения
Решетникова Валерия Николаевича,

доктора физико-математических наук, профессора, известного специалиста
в области информационных технологий, заместителя главного редактора
международного журнала «Программные продукты и системы».

Желаем здоровья, успехов, творческих свершений!

Надеемся, что наше дальнейшее сотрудничество будет неизменным и плодотворным.

Редколлегия

ФИЛЬТР КАЛМАНА КАК МЕТОД ВТОРИЧНОЙ ОБРАБОТКИ ИНФОРМАЦИИ С СИСТЕМЫ ГЛОНАСС

И.П. Болодурина, д.т.н., профессор, зав. кафедрой, prmat@mail.osu.ru;

А.А. Нугуманова, аспирант, alaymir@mail.ru

(Оренбургский государственный университет, просп. Победы, 13, г. Оренбург, 460018, Россия);

В.Н. Решетников, д.ф.-м.н., профессор, rvn_@mail.ru

(ФНЦ НИИСИ РАН, Нахимовский просп., 36, корп. 1, г. Москва, 117218, Россия)

В работе рассмотрены методы местоопределения наземного транспортного средства на основе вторичной обработки информации, поступающей со спутниковой навигационной системы ГЛОНАСС. Точность позиционирования транспортного средства, зависящая от многих факторов, в том числе и от метода нахождения координат, является одним из показателей эффективного управления транспортным средством, что позволяет оперативно контролировать передвижение транспортных средств, фактическое время работы транспортного средства и точность определения расхода топлива. В публикациях, посвященных проблеме местоопределения динамических объектов, как правило, не описываются математические модели и алгоритмы, а приводятся лишь конечные результаты. При этом практически отсутствуют источники, в которых была бы описана четкая алгоритмическая схема, пригодная для написания ПО. Во многом это связано с тем, что данное ПО является либо коммерческой тайной, либо интеллектуальной собственностью. В отличие от имеющихся методов контроля и мониторинга мобильных объектов авторами разработан пошаговый алгоритм фильтра Калмана, позволяющий увеличить точность позиционирования наземного транспортного средства. На основе предложенного алгоритма создано прикладное ПО, обеспечивающее в реальном масштабе времени и в режиме постобработки определение местоположения наземного объекта. Исследованы случаи подвижного и неподвижного состояний наземного транспортного средства, для каждого из которых применены алгоритм фильтра Калмана и метод наименьших квадратов, способствующие увеличению точности позиционирования. Приведена сравнительная характеристика указанных методов на основе выбранных статистических критериев и установлен наиболее точный метод местоопределения подвижного и неподвижного объектов.

Ключевые слова: местоопределение, ГЛОНАСС, транспортное средство, фильтр Калмана, метод наименьших квадратов.

Одной из наиболее важных проблем в сфере контроля и мониторинга мобильных объектов является их местоопределение. Погрешность в точность позиционирования транспортного средства вносят такие факторы, как атмосфера, возвышенные объекты, горы и др. [1]. Для определения точных координат различных объектов, скорости их движения и других факторов служит современная система глобальной спутниковой навигации ГЛОНАСС. Хотя использование ГЛОНАСС весьма актуально, анализ показывает, что технология позиционирования и идентификации мобильных объектов на пространственных цифровых моделях в транспортной сфере развита недостаточно. В публикациях, посвященных проблеме местоопределения, как правило, не описываются математические модели и алгоритмы, а приводятся лишь конечные результаты (например [2]). При этом практически отсутствуют источники с описанием четкой алгоритмической схемы, пригодной для написания ПО. Поэтому разработка инструментария для точного местоопределения наземных транспортных средств является актуальной задачей.

Методы вторичной обработки информации

В радиолокационных станциях автоматического сопровождения и обработки сигналов в инерциальных навигационных системах в качестве алгоритма вторичной обработки наиболее пригоден

фильтр Калмана [3–5]. Алгоритм работает в два этапа. На первом этапе фильтр Калмана экстраполирует значения переменных состояния, а также их неопределенности, на втором – по данным измерения уточняется результат экстраполяции. Благодаря пошаговой природе алгоритма он может в реальном времени отслеживать состояние объекта (без заглядывания вперед, используя только текущие замеры и информацию о предыдущем состоянии и его неопределенности) [6, 7].

Фильтр Калмана основан на квадратичном критерии качества [4, 8, 9], то есть обеспечивает минимум среднеквадратических ошибок оценок. Оценки определяются на основе условных математических ожиданий вектора состояния объекта наблюдения. Фильтр Калмана является рекурсивным методом, оперирует оценками не только состояния, но и неопределенности вектора состояния, опираясь на формулу Байеса условной вероятности:

$$f(\tilde{x}) = \frac{f(\hat{x} | \tilde{x}) \cdot f(\tilde{x})}{f(\tilde{x} | \hat{x})}, \quad (1)$$

где $f(\hat{x})$ – вероятность наступления апостериорной оценки; $f(\tilde{x})$ – вероятность априорной оценки; $f(\hat{x} | \tilde{x})$ – вероятность получения апостериорной оценки при выполнении априорной оценки; $f(\tilde{x} | \hat{x})$ – вероятность получения априорной оценки при выполнении апостериорной оценки.

По формуле Байеса можно более точно пересчитать вероятность, учитывая как ранее известную информацию, так и новые данные [7].

Рассмотрим постановку задачи в дискретном случае. Пусть объект и измерение описываются уравнениями

$$\hat{x}_{i+1} = A_i \hat{x}_i + z_{i+1}, \quad \hat{x}_{j0} = x_j, \quad j = \overline{1, N}, \quad (2)$$

$$z_{i+1} = C_i \bar{x}_{i+1} + v_i,$$

где x – исходные данные; z – измерения, полученные с помощью прогнозирования; $\hat{x}$ – апостериорная оценка, полученная с помощью фильтрации; $\bar{x}$ – априорная оценка; A – матрица эволюции процесса, $A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$; C – матрица наблюдений,

$C = (1 \ 0)_{m \times n}$; v_i – белый шум с интенсивностью R_0 .

Требуется, используя измеренные значения переменной z_i , найти несмещенную оценку $\hat{x}_i$ вектора x_i , обеспечивающего минимум квадрата ошибки ($e = x - \hat{x}$):

$$J = M[e_i^T e_i] \rightarrow \min, \quad (3)$$

где оценка $\hat{x}_i$ является несмещенной, если $Me_i = 0$.

Решение задачи определяется следующим образом [2, 7, 10]:

$$\hat{x}_i = x_i + K_i (z_i - C_i \bar{x}_i); \quad (4)$$

$$\bar{x}_{i+1} = A_i \hat{x}_i; \quad \bar{x}(i_0) = \bar{x}^0; \quad (5)$$

$$K_i = \tilde{P}_i C_i^T [C_i \tilde{P}_i C_i^T + R_0]^{-1}; \quad (6)$$

$$P_i = [E - K_i C_i] \tilde{P}_i [E - K_i C_i]^T + K_i R_0 K_i^T = \\ = \tilde{P}_i - \tilde{P}_i C_i^T [C_i \tilde{P}_i C_i^T + R_0]^{-1} C_i \tilde{P}_i; \quad (7)$$

$$\tilde{P}_{i+1} = A_i P_i A_i^T + Q_0^i; \quad (8)$$

$$\tilde{P}(i_0) = P_0. \quad (9)$$

Здесь $\bar{x}_i$ является математическим ожиданием вектора x_i и служит его априорной оценкой, то есть оценкой, которая получается до измерения z_i ; условие R_0^i означает, что ни одна координата выходной переменной не измеряется точно; $\tilde{P}_i$ – дисперсионная матрица ошибки $\tilde{e}_i = x_i - \bar{x}_i$, то есть ошибки априорной оценки; P_i – дисперсионная матрица ошибки $e_i = x_i - \hat{x}_i$, то есть ошибки искомой (апостериорной) оценки.

Кроме априорной оценки, выражение для оценки содержит поправочный член, пропорциональный невязке – разности между измеренным значением переменной z_i и ее оценкой.

При вычислении оценки каждый шаг нужно начинать с определения априорной дисперсионной матрицы $\tilde{P}_i$. Далее вычислить матрицу коэффициентов усиления K_i , дисперсионную матрицу P_i , затем априорную оценку $\bar{x}_i$ и в последнюю очередь – искомую оценку $\hat{x}_i$. Так как матрицы $\tilde{P}_i$, P_i , K_i не зависят от измерений, их можно вычислить заранее

при всех необходимых значениях i [10].

Допустим, что в момент времени t_i получена оценка вектора состояния системы $\hat{x}_i$ и необходимо получить оценку в момент t_{i+1} . Для этого строим прогноз априорной оценки $\bar{x}_{i+1}$ в момент t_{i+1} , опираясь на оценку вектора состояния системы $\hat{x}_i$, получаем измерения z_{i+1} , далее корректируем оценку в момент t_{i+1} , базируясь на прогнозе и измерениях, и получаем окончательную оценку вектора состояния $\hat{x}_{i+1}$ (рис. 1).

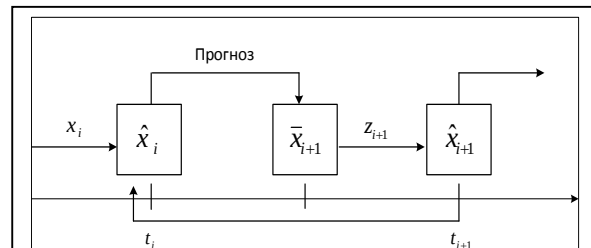


Рис. 1. Принцип работы фильтра Калмана

Fig. 1. How Kalman filter works

Разработанный алгоритм фильтра Калмана

Алгоритм фильтра Калмана включает следующие шаги:

1) формирование априорной матрицы $\bar{x}_i$, апостериорной матрицы $\hat{x}_i$, заполненных нулями;

2) формирование дисперсионной матрицы P , в которой диагональ матрицы заполняется единицами, элементы ниже главной диагонали являются нулями, а элементы выше главной диагонали вычисляются по формуле $P_{ij} = \frac{1}{j-i}$;

3) формирование матрицы H – матрицы преобразования состояния системы, в которой первый столбец заполняется единицами, остальные элементы вычисляются по формуле $\varepsilon_{ij} = -\frac{f^j}{j}$ (f – порядок фильтра);

4) формирование матрицы наблюдений C , в которой первый элемент равен единице, остальные элементы равны нулю;

5) получение матрицы B – матрицы перехода между состояниями: $H \cdot H^T = B$;

6) нахождение обратной матрицы B^{-1} и преобразование с помощью LU -разложения;

7) формирование матрицы идентичности Y с элементами

$$y_i = K - (K + L_{ij} \cdot y_j);$$

$$\tilde{P}_{ik} = y_i - (U_{ij} \cdot \tilde{P}_{jk}) - K / U_{ii},$$

где K – усиление фильтра Калмана, которое вычисляется по формуле (6);

8) формирование матрицы N – матрицы измерений $N_i = \frac{K^i}{i}$;

9) пересчет матрицы B : $B = N^T N + B$; повторяя шаги 7–9, получаем матрицу $\tilde{P}$;

10) расчет матрицы μ – матрицы определения шума процесса; $\mu = \tilde{P} \cdot C^T$;

11) пересчет $\mu_{ij} = \mu_{ij} \cdot (x - K)$, x – входные данные;

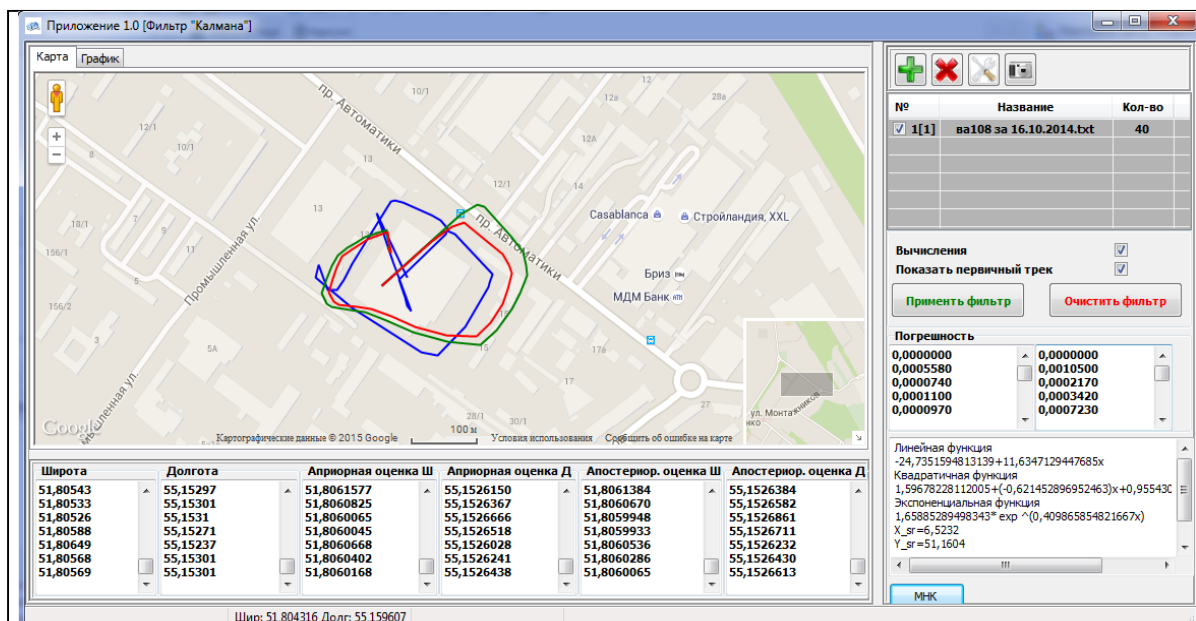
12) получение априорной оценки $\bar{x}_k + \mu = \hat{x}_k$;

13) получение апостериорной оценки $P \cdot \hat{x}_k = \bar{x}_k$.

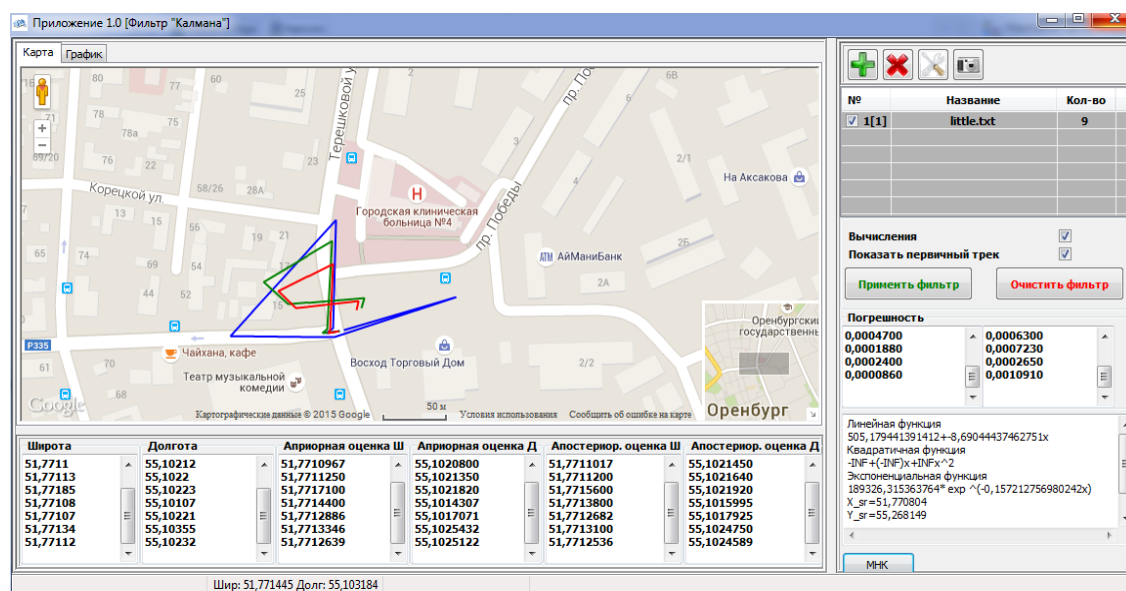
Программная реализация приведенного алгоритма фильтра Калмана позволяет получить координаты наземного транспортного средства с меньшими отклонениями.

Сравнительная характеристика

В данной работе проведено исследование для двух случаев: когда транспортное средство движется и когда находится на стоянке. Для оценки работы фильтра Калмана при обработке информации с системы ГЛОНАСС используем метод наименьших квадратов (МНК) [8, 9], где минимизируемый



а)



б)

Рис. 2. Экранная форма транспортного средства на карте: а) движущегося, б) неподвижного

Fig. 2. The screen shot of a vehicle on the map: а) moving, б) stationary

функционал имеет вид

$$J = \min(e^T e). \quad (10)$$

Рассмотрим задачу точного местоопределения координат транспортного средства на основе вторичной обработки информации спутниковой навигационной системы ГЛОНАСС алгоритмами фильтра Калмана и МНК.

Информационную базу составляют данные о местоположении маршрутных автобусов города Оренбурга, полученные с помощью программы «АТ Наблюдатель».

На основе приведенного алгоритма разработано ПО [11], отображающее траекторию движения транспортного средства (рис. 2), зафиксированную программой «АТ Наблюдатель» (синий цвет), априорную оценку траектории (красный цвет) и траекторию, полученную в результате применения фильтра Калмана (зеленый цвет). На рисунке 3 приведены траектории движения транспортного средства с указанием координат его местоположения. Проведенная проверка исходных данных на нормальный закон распределения позволяет оценить эффективность методов по различным критериям: средняя ошибка аппроксимации, коэффициент корреляции и критерий Фишера.

Полученные значения используемых статистических критериев, приведенные в таблице, позволяют сделать вывод о том, что для определения координат подвижного транспортного средства можно использовать оба метода. Тем не менее, результат обработки информации на основе фильтра Калмана является более точным с точки зрения допустимых предельных значений названных критериев.

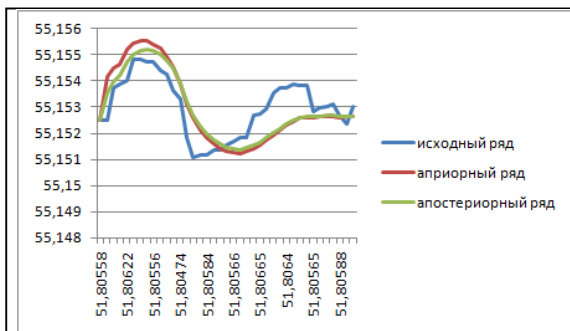
Критерии оценки работы методов

Methods evaluation criteria

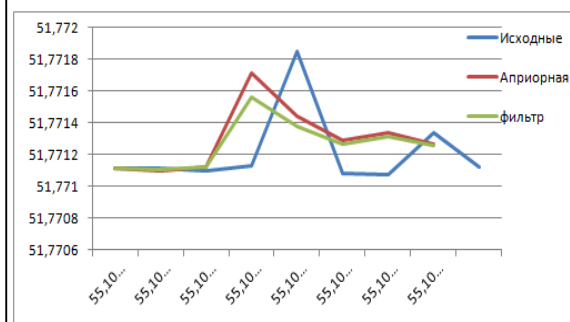
Критерий	Фильтр Калмана		МНК	
	П	Н	П	Н
Средняя ошибка аппроксимации, %	7,2	6,87	8,5	9,2
Коэффициент корреляции	0,753	0,885	0,49	0,426
F (факт.)	48,25	31,9	12,6	1,99
F (табл.)	4,12		5,32	

Примечание: П – подвижное транспортное средство, Н – неподвижное транспортное средство.

Проведена проверка эффективности фильтра Калмана и МНК для задачи местоопределения неподвижных объектов, то есть транспортных средств, находящихся на стоянке. Неподвижному транспортному средству должна соответствовать окрестность без резких скачков, но из-за факторов, влияющих на точность (атмосфера, высокие здания и так далее), программа «АТ Наблюдатель» фиксирует траекторию, соответствующую движущемуся объекту в некоторой окрестности точки расположения, что является неприемлемым с экономиче-



а)



б)

Рис. 3. График для транспортного средства: а) движущегося, б) неподвижного

Fig. 3. A vehicle graph: а) moving, б) stationary

ской точки зрения (расход топлива, увеличение пробега транспортного средства и т.д.).

Вычисленные значения критериев в случае корректирования местоположения неподвижного транспортного средства, приведенные в таблице, позволяют сделать вывод о том, что фильтр Калмана является более точным для определения координат. МНК не удовлетворяет допустимым пределам выбранных критериев, в частности, предельным значениям критерия Фишера.

В заключение отметим, что разработанный алгоритм фильтра Калмана позволит решить проблему позиционирования наземных транспортных средств. В результате проведенной сравнительной характеристики установлено, что предпочтительным методом для корректирования местоположения подвижных и неподвижных наземных транспортных средств на основе вторичной обработки информации, поступающей со спутниковой навигационной системы ГЛОНАСС, является фильтр Калмана.

Литература

1. Решетников В.Н. Космические телекоммуникации. Системы спутниковой связи и навигации. СПб: Ленинград. изд-во, 2010. 134 с.
2. Манухов В.Ф., Разумов О.С., Спиридонов А.И., Тюряхин А.С. Спутниковые методы определения координат пунктов геодезических сетей: учеб. пособие. Саранск: Изд-во Мордов. ун-та, 2009. 108 с.
3. Болодурина И.П., Решетников В.Н., Таспаева М.Г. Ме-

тоды уточнения интегрированного наземно-космического мониторинга в системе ГЛОНАСС // Программные продукты и системы. 2011. № 4. С. 130–134.

4. Калман Р., Фалб П., Арбиб М. Очерки по математической теории систем: учеб. пособие; [пер. с англ.]. М.: Мир, 1971. 400 с.

5. Сучилин В.И., Волобуев Г.Б. Оценка возможностей повышения точности местоопределения наземного подвижного объекта путем вторичной обработки показаний аппаратуры пользователя систем GPS NAVSTAR и/или ГЛОНАСС // Кибернетика и высокие технологии XXI века: сб. докл. VIII Междунар. науч.-технич. конф. Воронеж, 2007. Т. 2. С. 1066–1073.

6. Sigurd I. Aanonsen. The Ensemble Kalman Filter in Reservoir Engineering – a Review / Sigurd I. Aanonsen, Dean S. Oliver, Albert C. Reynolds, Brice Valles. Intern. Research Institute of Stavanger, SPE Journ., 2009, no. 14, pp. 393–412.

7. Welch G., Bishop G. An Introduction to the Kalman Filter / Welch G. / UNC – Chapel Hill, TR 95-041, 2006; URL: http://cs.unc.edu/~welch/media/pdf/kalman_intro (дата обращения: 30.08.2015).

8. Сергиенко А.Б. Цифровая обработка информации. СПб: БХВ–Петербург, 2011. 768 с.

9. Тихонов В.И., Шахтарин Б.И., Сизых В.В. Случайные процессы. М.: Горячая линия–Телеком, 2009. 399 с.

10. Яценков В.С. Основы спутниковой навигации. Системы GPS NAVSTAR и ГЛОНАСС. М.: Горячая линия–Телеком, 2005. 272 с.

11. Болодурина И.П., Нугуманова А.А., Курочкина Т.А. Вторичная обработка информации для корректировки местоположения наземного транспортного средства методом фильтра Калмана. Свид. о регистр. прог. для ЭВМ № 50201550245. М.: ЦИТИС, 2015.

DOI: 10.15827/0236-235X.112.116-120

Received 09.09.15

KALMAN FILTER AS A METHOD OF SECONDARY PROCESSING OF GLONASS SYSTEM INFORMATION

Bolodurina I.P., Dr.Sc. (Engineering), Professor, Head of Department, prmat@mail.osu.ru;

Nugumanova A.A., Postgraduate Student, alaymir@mail.ru

(Orenburg State University, Pobedy Ave. 13, Orenburg, 460018, Russian Federation);

Reshetnikov V.N., Dr.Sc. (Physics), Professor, rvn_@mail.ru

(SRISA RAS, Nakhimovsky Ave. 36/1, Moscow, 117218, Russian Federation)

Abstract. The paper discusses methods of positioning a land vehicle based on secondary processing of information from GLONASS satellite navigation system. Vehicle positioning accuracy depends on many factors including the method of finding the coordinates. It is one of the indicators of vehicle effective control. Therefore, one can quickly control the vehicles movement, their actual value and the fuel consumption accuracy. The publications devoted to the problem of dynamic objects positioning, as a rule, do not describe mathematical models and algorithms, they just display the final results. In fact there are no sources, which can clearly describe an algorithmic scheme suitable for developing software. This is due to the fact that the software is a trade secret or an intellectual property. In contrast to existing methods of mobile objects control and monitoring, the authors have developed a Kalman filter algorithm step by step. It increases the positioning accuracy of a land vehicle. The algorithm is a basis for application software, which determines the land object location in real-time and post-processing mode. The paper discusses the cases of mobile and stationary state of a land vehicle. Each case shows using Kalman filter algorithm and the least squares method to enhance positioning accuracy. There is a comparative characteristics of these methods on the basis of selected statistical criteria. The authors set the most accurate method of positioning mobile and stationary objects.

Keywords: positioning, GLONASS, vehicle, Kalman filter, method of least squares.

References

1. Reshetnikov V.N. *Kosmicheskie telekommunikatsii. Sistemy sputnikovoy svyazi i navigatsii* [Space Telecommunications. Satellite Communication and Navigation Systems]. St.-Petersburg, Leningrad Publ., 2010, 134 p.

2. Manukhov V.F., Razumov O.S., Spiridonov A.I., Tyuryakhin A.S. *Sputnikovye metody opredeleniya koordinat punktov geodezicheskikh setey* [Satellite methods to determine the coordinates of geodetic network points]. Study guide, Saransk, Mordovia Univ. Publ., 2009, 108 p.

3. Bolodurina I.P., Reshetnikov V.N., Taspavaeva M.G. Verifying Methods for Integrated Ground-Space Monitoring in GLONASS. *Programmnye produkty i sistemy* [Software & Systems]. 2011, no. 4, pp. 130–134 (in Russ.).

4. Kalman R., Falb P., Arbib M. *Topics in mathematical system theory*. McGraw Hill Book Co., 1969 (Russ. ed.: E.L. Nappelbaum, Moscow, Mir Publ., 1971, 400 p.).

5. Suchilin V.I., Volobuev G.B. Assessment of opportunities to improve the accuracy of ground moving object positioning by secondary processing of GPS NAVSTAR and/or GLONASS readings. *Kibernetika i vysokie tekhnologii XXI veka: sb. dokl. VIII Mezhdunar. nauch.-tekhnich. konf.* [Proc. of the 8th Int. Science and Tech. Conf. "Computer Science and High Technologies of the XXI century"]. Voronezh, 2007, vol. 2, pp. 1066–1073 (in Russ.).

6. Aanonsen S.I., Oliver D.S., Reynolds A.C., Valles B. The Ensemble Kalman Filter in Reservoir Engineering – a Review. Intern. Research Institute of Stavanger. *SPE Journ.* 2009, no. 14, pp. 393–412.

7. Welch G., Bishop G. *An Introduction to the Kalman Filter*. UNC – Chapel Hill, TR 95-041, 2006 Available at: http://cs.unc.edu/~welch/media/pdf/kalman_intro (accessed August 30, 2015).

8. Sergienko A.B. *Tsifrovaya obrabotka informatsii* [Digital Information Processing]. St.-Petersburg, BKhV–Peterburg Publ., 2011, 768 p.

9. Tikhonov V.I., Shakhhtar B.I., Sizykh V.V. *Sluchaynye protsessy* [Stochastic Processes]. Moscow, Goryachaya liniya–Telecom Publ., 2009, 399 p.

10. Yatsenkov V.S. *Osnovy sputnikovoy navigatsii. Sistemy GPS NAVSTAR i GLONASS* [Fundamentals of Satellite Navigation. GPS NAVSTAR and GLONASS]. Moscow, Goryachaya liniya–Telecom Publ., 2005, 272 p.

11. Bolodurina I.P., Nugumanova A.A., Kurochikina T.A. *Vtorichnaya obrabotka informatsii dlya korrektyrovki mestopolozheniya nazemnogo transportnogo sredstva metodom filtra Kalmana* [Data Secondary Processing to Adjust the Location of a Land Vehicle by the Kalman Filter]. The Certificate of a PC program, registration no. 50201550245. Moscow, TsITiS Publ., 2015.

УДК 681.3:629.7

Дата подачи статьи: 06.10.15

DOI: 10.15827/0236-235X.112.121-125

ИНЪЕКТОР СБОЕВ ДЛЯ ТЕСТИРОВАНИЯ МИКРОПРОЦЕССОРОВ ТИПА СИСТЕМА НА КРИСТАЛЛЕ К ОДИНОЧНЫМ СБОЯМ

С.А. Чекмарёв, инженер, zaq259@yandex.ru

(Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнева,
просп. им. газ. «Красноярский рабочий», 31, г. Красноярск, 660014, Россия)

Для тестирования механизмов обнаружения и коррекции сбоев используют инъектирование ошибок в память микропроцессора. Инъектирование имитирует результат воздействия космических тяжелых заряженных частиц. В работе рассмотрен IP-блок инжектора сбоев. IP-блок вносит одиночные сбои в память микропроцессора типа система на кристалле. Приведены схема инжектора сбоев, состав модулей, машина состояний, кратко описана работа во всех режимах инъектирования. Инъектор позволяет вносить сбои в регистровый файл, кэш-память и внешнюю память в различных режимах. Доступны следующие режимы: режим с остановкой и без остановки процессора со случайным или предопределенным покрытием сбоем. IP-блок инжектора сбоев был применен в СнК-процессоре LEON3. В статье описаны процедуры инъектирования сбоев в память LEON3. Доступ к внутрикристалльной памяти осуществляется через DSU-интерфейс. DSU является ведомым устройством на AMBA АНВ-шине. Через его регистры инжектор сбоев может остановить работу процессора, модифицировать содержимое внутренней памяти, возобновить работу ПО. Для доступа к внешней памяти процессора LEON3 используется контроллер памяти. Во время эксперимента инжектор собирает статистическую информацию о внесенных и обнаруженных сбоях. Анализ результатов позволяет сделать вывод о чувствительности микропроцессора к одиночным сбоям в памяти. По завершении тестирования инжектор сбоев может быть исключен из состава микропроцессора типа система на кристалле, не оставив в нем никаких «следов».

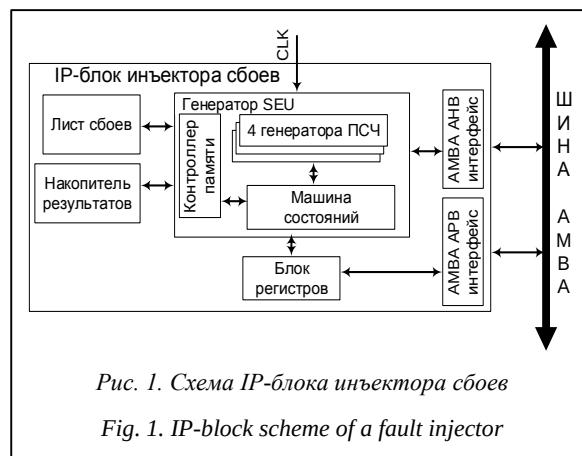
Ключевые слова: одиночные сбои, инъектирование, система на кристалле, LEON3, инжектор сбоев.

Обеспечение сбоеустойчивости микропроцессоров, работающих в составе бортовой космической аппаратуры, является важнейшей задачей электронного космического приборостроения. Среди других сбоев, присущих аппаратуре в космосе, своей частотой обращают на себя внимание сбои, вызванные событиями SEU (Single Event Upset) [1, 2]. Данные сбои происходят в памяти: как внутри микропроцессора (регистры и кэш-память), так и во внешней оперативной памяти. Для предотвращения данных сбоев используют помехоустойчивые коды (Error Correction Code, ECC) [3], позволяющие находить и автоматически исправлять одну-две ошибки в памяти. Процессоры и созданные на их основе микропроцессорные системы, которые имеют механизмы обнаружения и исправления ошибок (Error Detection And Correction, EDAC) на базе ECC, называют сбоеустойчивыми.

Для тестирования сбоеустойчивости процессоров к SEU используют методы инъекции сбоев (Fault Injection, FI) [4–9], которые заключаются в искусственном инъектировании сбоев в память микропроцессора, эмулируя результат воздействия на цифровую аппаратуру тяжелых заряженных частиц космического ионизирующего излучения. В работах [10–12] представлен метод инъектирования сбоев в память микропроцессора с помощью блока инъекций сбоев, размещаемого в микропроцессоре типа система на кристалле (СнК) в качестве сложно-функционального блока (IP-блока, Intellectual Property). Данный блок позволяет проводить автономные от внешнего аппаратно-программного окружения инъекции как во внутреннюю, так и во внешнюю память. Сбои могут производиться с остановкой процессора перед проведением инъектирования, а также без остановки

процессора при инъектировании во внешнюю оперативную память. Покрытие сбоев может быть случайным равномерным или предопределенным.

В данной работе рассмотрен собственно сам IP-блок инжектора сбоев для СнК-процессора LEON3 [13], определен его состав, кратко описана его работа во всех режимах инъектирования. Схема IP-блока представлена на рисунке 1.



IP-блок инжектора сбоев состоит из шести частей (блоков). Блок «АМБА АНВ-интерфейс» согласует инжектор сбоев с шиной АМБА АНВ. При этом сам инжектор может выступать в роли как ведущего устройства на шине АМБА, так и ведомого. Режим ведущего устройства позволяет инициировать транзакции по АМБА-шине, необходимые для проведения инъекций сбоев в память микропроцессора. Режим ведомого устройства позволяет получить доступ к блокам «Лист сбоев» и «Накопитель результатов» другим устройствам на шине АМБА, что дает возможность вручную инициализировать

первый блок и прочитать второй, например, с помощью внешнего управляющего компьютера, подключенного к системе по UART-интерфейсу.

Запрос на инициализацию транзакции для проведения инъекции генерируется в блоке «Генератор SEU». Режим работы генератора SEU определяется соответствующей «Машиной состояний», управляемой с помощью блока регистров по AMBA APB-интерфейсу.

Четыре генератора *псевдослучайных чисел* (ПСЧ) обеспечивают возможность автоматической генерации следующих параметров, необходимых для инъекции:

- адрес ошибки во внутренней памяти;
- положение ошибки в 32-битном слове внутренней памяти;
- момент времени для инъекции ошибки (определяется интервалом между инъекциями, выраженным в системных тактах);
- вид ошибки: однократная (ошибка в одном бите) или двукратная (ошибка в двух соседних битах).

ПСЧ реализованы на базе линейного регистра сдвига с обратной связью Linear Feedback Shift Register, LFSR [14].

Генерируемые данные сохраняются в блоке «Лист сбоев», который представляет собой модуль внутрикристалльной 32-разрядной памяти. Для обеспечения доступа к данной памяти с различных устройств (ручной/автоматический режимы инициализации листа сбоев) реализован контроллер памяти, обрабатывающий возможные коллизии. Следует отметить, что режим ручного редактирования возможен только при условии, что генератор сбоев не находится в активном состоянии (до начала FI-компании). Данное условие позволило сократить аппаратные ресурсы при практической реализации контроллера.

Управление инжектором сбоев осуществляется с помощью «Блока регистров», доступного для чтения/записи по AMBA APB-интерфейсу. Регистры блока приведены в таблице 1.

Блок «Накопитель результатов» представляет собой модуль внутрикристалльной памяти разрядностью 32 бита. Он служит для накопления результатов проводимых FI-компаний.

Работа инжектора сбоев INJ_SEU определяется «Машиной состояний» (рис. 2), входящей в состав блока «Генератор сбоев», управление которой определяется блоком регистров. Конечный автомат состоит из девяти состояний: idle, gen_params, read_sheet, pending, snooping, stop_proc, injection, start_proc, statistics_collection. Idle – режим простоя. В этом режиме генератор сбоев неактивен. У оператора есть доступ на запись для управляющих регистров и листа сбоев. Таким образом, в этом состоянии можно провести первичную инициализацию контроллера. При активации инжектора (бит «en» управляющего регистра установлен в 1) гене-

ратор сбоев переходит из состояния Idle в состояние gen_params или read_sheet. Выбор определяется наличием/отсутствием флага «sh» управляющего регистра.

Таблица 1

Регистры IP-ядра инжектора сбоев

Table 1

IP-core registers of a fault injector

Смещение относительно базового адреса на шине APB	Регистр
0x00	Регистр управления
0x04	Регистр задержки
0x08	Регистр статистики кэш-инструкций (область тэга)
0x0c	Регистр статистики кэш-инструкций (область данных)
0x10	Регистр статистики кэш-данных (область тэга)
0x14	Регистр статистики кэш-данных (область данных)
0x18	Регистр статистики регистрового файла
0x1c	Регистр статистики внешнего ОЗУ
0x20	Регистр предопределенного адреса
0x24	Регистр предопределенных данных

Если лист сбоев не был инициализирован вручную, генератор сбоев сам сгенерирует необходимые параметры, используя для этого генераторы ПСЧ в состоянии gen_params. Следует отметить, что в этом случае нет возможности проведения инъекции в режиме работы «предопределенный». Далее конечный автомат перейдет в состояние pending, ожидая момента наступления инъекции, который определяется суммой задержек, выставленной в регистре Time и сгенерированной генератором ПСЧ. Затем, исходя из режима работы инжектора (с остановкой/без остановки процессора), конечный автомат перейдет в состояние stop_proc/injection соответственно. Режим работы инжектора определяется флагом SM управляющего регистра. Следует отметить, что в режиме «без остановки процессора» инъекции возможны только в ОЗУ процессора. В состоянии injection генератор производит операции чтения/записи данных в памяти. Адрес и положение сбоя в 32-битном слове в данном случае будут определяться сгенерированными параметрами в состоянии gen_params. По завершении процесса инъекции конечный автомат перейдет либо в состояние start_proc (если активирован соответствующий режим), либо напрямую в состояние statistics_collection. Состояние statistics_collection определяет операции чтения статусных регистров механизма EDAC тестируемого процессора. По завершении процесса сбора статистики конечный автомат перейдет в состояние gen_params и FI компания будет продолжена. По завер-

шении необходимого числа инъекций ($N_{seu} = \max$) конечный автомат сбросит флаг EN в 0 и вернется в состояние Idle. Значение $\max$ определяется размером листа сбоев инжектора.

Если лист сбоев был инициализирован вручную, конечный автомат перейдет в состояние `read_sheet` с целью чтения необходимых параметров. Исходя из режима работы инжектора сбоев (определяется флагом РМ управляющего регистра), конечный автомат перейдет в состояние либо `pending` (и его работа будет соответствовать описанному выше алгоритму), либо `snooring`. В состоянии `snooring` конечный автомат будет мониторить моменты обращения во внешнюю память процессора. При достижении необходимого `watchpoint` либо будет остановлен процессор в состоянии `stop_proc` (флаг SM управляющего регистра имеет значение 1), либо при отсутствии обращений к контроллеру памяти будет произведена инъекция в ОЗУ в состоянии `injection`. Сбрав статистику обнаруженных ошибок в состоянии `statistics_collection`, конечный автомат вновь перейдет в состояние `read_sheet`. При достижении конца листа сбоев конечный автомат сбросит флаг EN в 0 и вернется в состояние `Idle`.

Чтение/запись данных в память процессора LEON3 производится с внутрикристального отладчика (DSU). DSU является ведомым устройством на AMBA AHB-шине. Для взаимодействия с процессором LEON3 инжектор по DMA-каналу обращается к регистрам DSU. Используется набор регистров отладочного интерфейса, представленный в таблице 2.

Например, для остановки процессора LEON3 необходимо установить 1 в бит Break now (BN0) регистра Break and Single Step. Для этого в состоянии stop_proc производится операция записи по адресу DSU_start_address+0x000020 слова 0x1. Аналогично производится операция запуска процессора – путем сбрасывания данного бита в 0 в состоянии start_proc.

Для выполнения инъекций в регистровый файл необходимо отключить механизм EDAC. Для этого проводится операция записи слова 0x1 по адресу

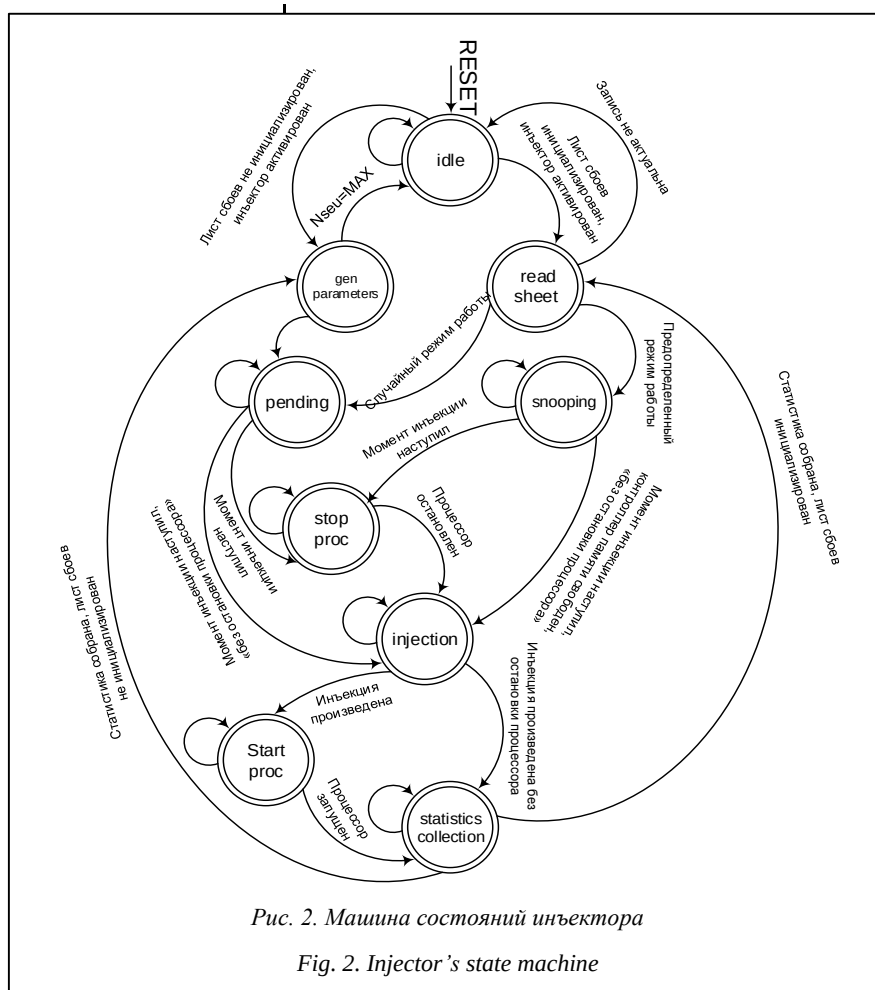


Рис. 2. Машина состояний иньектора

Fig. 2. Injector's state machine

DSU_start_address+0x400040 (регистр asr16). Далее иньектор читает/модифицирует содержимое регистрового файла, используя адрес DSU_start_address+0x300000+error_address. Описанные операции выполняются в состоянии injection.

Таблица 2

Карта памяти DSU-интерфейса, используемая инжектором сбоев

Table 2

DSU interface memory card, which is used by a fault injector

Смещение относительно базового адреса на шине APB	Регистр
0x00	DSU control register
0x000020	Break and Single Step register
0x300000– 0x3007FC	IU register file
0x300800– 0x300FFC	IU register file check bits (LEON3FT only)
0x400024	DSU ASI register
0x400040	ASR16
0x700000– 0x7FFFFC	ASI diagnostic access (ASI = value in DSU ASI register, address = address[19:0])

Аналогично инжектор сбоев модифицирует содержимое кэш-памяти. При этом используется диагностический регистр интерфейса DSU (ASI). Инжектор по DMA-каналу записывает в него ($DSU_start_address + 0x400024$) следующие значения: $0x9$ для доступа к кэшу инструкций (область данных), $0xB$ для доступа к кэшу данных (область данных), $0xC$ для доступа к кэшу инструкций (область тэга), $0xE$ для доступа к кэшу данных (область тэга). Далее производятся операции чтения/модификации по адресу $DSU_start_address + 0x700000 + error_address$.

Доступ к внешнему ОЗУ осуществляется без DSU, напрямую через контроллер памяти. До осуществления инъекции инжектор сбоев отключает механизм EDAC контроллера памяти путем установки в единицу бита RE (RAM EDAC enable). Далее осуществляется операция чтения/модификации по адресу $ram_start_address + error_address$.

Во всех случаях при осуществлении инъекций модифицированные данные определяются следующим образом: $D_{mod} = D_{read} \text{ xor } Error_mask$, где $Error_mask$ – параметр, определяемый на стадии $read_sheet$ или $gen_parameters$; D_{read} – содержимое памяти; D_{mod} – данные, записанные в результате инъекции.

Инжектор сбоев является и ведомым устройством на шине AMBA AHB. Эта функция позволяет получать доступ к листу сбоев и накопителю результатов инжектора для других устройств, входящих в состав СнК. Одним из входных сигналов инжектора является $ahbsi : in_ahb_slv_in_type$. Рассмотрим его состав.

```
-- AHB slave inputs
type ahb_slv_in_type is record
  hsel   : std_logic_vector(0 to NAHBSLV-1); -- slave select
  haddr  : std_logic_vector(31 downto 0);    -- address bus (byte)
  hwrite : std_ulogic;                       -- read/write
  htrans : std_logic_vector(1 downto 0);     -- transfer type
  hsize  : std_logic_vector(2 downto 0);     -- transfer size
  hburst : std_logic_vector(2 downto 0);     -- burst type
  hwdata : std_logic_vector(31 downto 0);     -- write data bus
  hprot  : std_logic_vector(3 downto 0);     -- protection control
  hready : std_ulogic;                       -- transfer done
  hmaster : std_logic_vector(3 downto 0);    -- current master
  hmasterlock : std_ulogic;                 -- locked access
  hmbssel : std_logic_vector(0 to NAHBAMR-1); -- memory bank select
  hcache  : std_ulogic;                     -- cacheable
  hirq    : std_logic_vector(NAHBIRQ-1 downto 0); -- interrupt result bus
  testen  : std_ulogic;                     -- scan test enable
  testrst : std_ulogic;                     -- scan test reset
  scanen  : std_ulogic;                     -- scan enable
  testoen : std_ulogic;                     -- test output enable
end record;
```

Сигналы $hselect$, $hwrite$, $haddress$, $hmaster$ позволяют инжектору контролировать шину адреса и данных, а также направление передачи (write/read). Эта особенность позволяет инжектору осуществлять мониторинг транзакций на шине AMBA AHB. Данный режим используется в состоянии snooping, когда инжектор работает в привилегированном режиме и ожидает наступления определенного события (например, запись определенных данных по конкретному адресу оперативной памяти). Как только это событие происходит, осуществляется инъекция.

При работе в предопределенном режиме с остановкой процессора инжектор может производить инъекции и в другие виды памяти. Для этого можно установить специальные контрольные точки (watchpoint) в регистре процессора. LEON3 поддерживает до четырех подобных контрольных точек с возможностью наложения на них устанавливаемых масок. Таким образом, до проведения FI-компаний оператор конфигурирует соответствующим образом регистры процессора и инжектор сбоев. При срабатывании watchpoint процессор сам перейдет в режим отладки и будет остановлен. Инжектор может определить наступление этого события по установленному биту DM (Debug mode in control register) DSU-интерфейса. Ожидание наступления данного события также происходит в состоянии snooping. Далее будет произведена инъекция путем модификации содержимого интересующей области памяти.

Таким образом, разработанный инжектор сбоев позволяет вносить сбои в регистровый файл, кэш-память и внешнюю память в режимах с остановкой и без остановки процессора со случайным или предопределенным покрытием сбоями. Являясь независимым IP-блоком, он имеет слабую связь с остальными блоками LEON3, что позволяет проводить его простое включение/выключение из состава данного СнК-процессора.

Литература

- Осипенко П. Одиночные сбои – вызов современных микропроцессоров // Электронные компоненты. 2009. № 7. С. 12–15.
- Максименко С.Л., Мелехин В.Ф., Филиппов А.С. Анализ проблемы построения радиационно-стойких информационно-управляющих систем // Информационно-управляющие системы. 2012. № 2. С. 18–25.
- Золотарев В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы: справочник. М.: Горячая линия–Телеком, 2004. 126 с.
- Yuste P., de Andres D., Lemus L., Serrano J., Gil P. Inerte: Integrated Nexus-Based Real-Time Fault Injection Tool for Embedded Systems. Intern. Conf. on Dependable Systems and Networks. SF, CA, 2003, p. 669.
- Zenha-Rela M., Cunha J.C., Santos L.E., Gameiro M., Gonçalves P., Alves G., Fidalgo A., Fortuna P., Maia R., Henriques L., Costa D. Exploiting the IEEE 1149.1 Standard for Software Reliability Evaluation in Space Applications. European Safety and Reliability Conf., 2006, pp. 1459–1464.
- Sonza R.M., Sterpone L., Violante M., Portela-Garcia M., Lopez-Ongil C., Entrena L. Fault Injectionbased Reliability Evaluation of SoPCs. 11th IEEE European Test Sympos., 2006, pp. 75–82.
- Portela-Garcia M., Lopez-Ongil C., Garcia-Valderas M., Entrena L. Fault Injection Approach for Measuring SEU Sensitivity in Complex Processors. A Rapid. 13th IEEE Intern. Online Testing Sympos. Heraklion, Crete, Greece July 8–11, 2007, pp. 101–106.
- Fidalgo A., Gerigota M., Alves G., Ferreira J. Real-time fault injection using enhanced on-chip debug infrastructures. Microprocessors and Microsystems, 2011, no. 25, pp. 441–452.
- Fidalgo A., Alves G., Ferreira J. A modified debugging infrastructure to assist real time fault injection campaigns. 9th IEEE Workshop on Design & Diagnostics of Electronic Circuits & Systems, 2006, pp. 172–177.
- Чекмарев С.А. Способ и система инъекции ошибок для тестирования сбоеустойчивых процессоров бортовых систем космических аппаратов // Вестн. СибГАУ. 2014. № 4 (56). С. 132–138.

11. Chekmarev S.A., Khanov V.Kh., Antamoshkin O.A. Modification of Fault Injection Method via On-Chip Debugging for Processor Cores of Systems-On-Chip. *Proc. Intern. Siberian Conf. on Control and Communications (SibCon)*, Russia, Omsk, 2015, pp. 1–4.

12. Чекарчев С.А. Инъекция сбоев в процессорные ядра систем на кристалле методом внутрикristальной отладки в реальном времени // Современные проблемы радиоэлектроники: сб. науч. тр. Красноярск: Сиб. федер. ун-т, 2015. С. 235–239; URL:

http://efir.sfu-kras.ru/wp-content/uploads/download/Сборник_СПП-2015.pdf (дата обращения: 05.10.2015).

13. LEON3 Processor // Aeroflex Gaisler 2015. URL: <http://www.gaisler.com/index.php/products/processors/leon3> (дата обращения: 05.10.2015).

14. Linear Feedback Shift Registers // New Wave Instruments 2015. URL: http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm (дата обращения: 05.10.2015).

DOI: 10.15827/0236-235X.112.121-125

Received 06.10.15

A FAULT INJECTOR TO TEST THE SoC-PROCESSOR TO SINGLE EVENT UPSETS

Chekmarev S.A., Engineer, zaq259@yandex.ru

(Academician M.F. Reshetnev Siberian State Aerospace University,
Krasnoyarsky Rabochy Ave. 31, Krasnoyarsk, 660014, Russian Federation)

Abstract. Error injection to the microprocessor memory is used to test fault detection and correction tools. Injection simulates the result of space heavy charged particles influence. This paper describes the developed failure injector IP-core. IP-core makes single event upsets to the memory of the system-on-a-chip (SoC) microprocessor. The paper describes a failure injector scheme, modules structure, a state machine, the work in all injection modes. The injector allows us to add failures to a register file, cache and external memory in different modes. Available modes are: with or without a microprocessor stop the processor with a random failure or predetermined by covering. Failure injector IP-core was applied to the SoC processor LEON3. The article describes the failure injection procedures in LEON3 memory. DSU interface helps to access the on-chip memory. DSU is a slave interface to the AMBA AHB bus. Using its registers the failure injector can stop the processor, modify its internal memory contents, and resume running software. A memory controller is used to access LEON3 external memory. During the experiment the injector collects statistical data on injected and found faults. The analysis of the results leads to a conclusion about the microprocessor sensitivity to single event upsets in the memory. After testing a failure injector can be excluded from the SoC microprocessor without any traces left.

Keywords: single event upset, injection, system-on-chip, LEON3, SEU injector.

References

1. Osipenko P. Single failures are a challenge to modern microprocessors. *Elektronnye komponenty* [Electronic Components]. 2009, no. 7, pp. 12–15 (in Russ.).
2. Maksimenko S.L., Melekhin V.F., Filippov A.S. Analysis of the problem of building radiation-resistant information-management systems. *Informatsionno-upravlyayushchie sistemy* [Information and Control Systems]. 2012, no. 2, pp. 18–25 (in Russ.).
3. Zolotarev V.V., Ovechkin G.V. *Pomekhoustoychivoe kodirovaniye. Metody i algoritmy* [Error-correction coding. Methods and algorithms]. 2004, Moscow, Goryachaya liniya–Telekom Publ., 126 p.
4. Yuste P., de Andres D., Lemus L., Serrano J., Gil P. Inerte: Integrated Nexus-Based Real-Time Fault Injection Tool for Embedded Systems. *Intern. Conf. on Dependable Systems and Networks*. SF, CA, 2003, p. 669.
5. Zenha-Rela M., Cunha J.C., Santos L.E., Gameiro M., Goncalves P., Alves G., Fidalgo A., Fortuna P., Maia R., Henriques L., Costa D. Exploiting the IEEE 1149.1 Standard for Software Reliability Evaluation in Space Applications. *European Safety and Reliability Conf.* 2006, pp. 1459–1464.
6. Sonza R.M., Sterpone L., Violante M., Portela-García M., Lopez-Ongil C., Entrena L. Fault Injectionbased Reliability Evaluation of SoPCs. *Proc. of the 11th IEEE European Test Symp.* 2006, pp. 75–82.
7. Portela-Garcia M., Lopez-Ongil C., Garcia-Valderas M., Entrena L. Fault Injection Approach for Measuring SEU Sensitivity in Complex Processors. *A Rapid. 13th IEEE Int. Online Testing Symp.* Heraklion, Crete, Greece, 2007, pp. 101–106.
8. Fidalgo A., Gerigota M., Alves G., Ferreira J. Real-time fault injection using enhanced on-chip debug infrastructures. *Microprocessors and Microsystems*. 2011, no. 25, pp. 441–452.
9. Fidalgo A., Alves G., Ferreira J. A modified debugging infrastructure to assist real time fault injection campaigns. *Proc. of the 9th IEEE Workshop on Design & Diagnostics of Electronic Circuits & Systems*. 2006, pp. 172–177.
10. Chekmarev S.A. Method and system of error injection for testing the fault tolerance processor on-the board control system of spacecraft. *Vestnik SibGAU* [The Bulletin of SibSAU]. 2014, no. 254(56), pp. 132–138 (in Russ.).
11. Chekmarev S.A., Khanov V.Kh., Antamoshkin O.A. Modification of Fault Injection Method via On-Chip Debugging for Processor Cores of Systems-On-Chip. *Proc. of the 2015 Int. Siberian Conf. on Control and Communications (SibCon)*. Omsk, 2015, pp. 1–4 (in Russ.).
12. Chekmarev S.A. Fault injection in systems-on-chip processors using the real-time on-chip debugging method. “Sovremennye problemy radioelektroniki”, sb. nauch. trudov [“Modern problems in radioelectronics”, Proc.]. Krasnoyarsk, Siberian Fed. Univ. Publ., 2015, pp. 235–239. Available at: http://efir.sfu-kras.ru/wp-content/uploads/download/Сборник_СПП-2015.pdf (accessed September 10, 2015).
13. LEON3 Processor. Available at: <http://www.gaisler.com/index.php/products/processors/leon3> (accessed October 05, 2015).
14. Linear Feedback Shift Registers. Available at: http://www.newwaveinstruments.com/resources/articles/m_sequence_linear_feedback_shift_register_lfsr.htm (accessed September 10, 2015).

УДК 004.78:005.7(075.8); 004.9:681.5; 621.3.068
DOI: 10.15827/0236-235X.112.126-132

Дата подачи статьи: 27.07.15

ПОДХОД К РАЗВИТИЮ СИСТЕМЫ УПРАВЛЕНИЯ ТЕСТИРОВАНИЕМ ПРОГРАММНЫХ СРЕДСТВ

В.Ф. Корнюшко, д.т.н., профессор, зав. кафедрой

*(Московский государственный университет тонких химических технологий им. М.В. Ломоносова,
просп. Вернадского, 86, г. Москва, 119571, Россия);*

А.В. Костров, д.т.н., профессор, akostrov@rambler.ru;

П.А. Породникова, аспирант, polina.porodnikova@mail.ru

*(Владимирский государственный университет им. Александра Григорьевича
и Николая Григорьевича Столетовых, ул. Горького, 87, г. Владимир, 600000, Россия)*

В статье поставлена задача формирования подхода к управлению уровнем развития *системы управления тестированием* (СУТ) в составе системы управления бизнес-процессами разработки ПО в условиях проектного предприятия. Предложено выделить в составе бизнес-процессов разработки ПО бизнес-процесс тестирования, а в составе системы управления разработкой – подсистему управления тестированием как самостоятельную. Рассмотрены особенности организации тестирования в типовых моделях разработки ПО, для различных моделей разработок построены варианты организации выполнения по этапам основных процессов: рецензирование, Review (R); разработка тестов, Test Design (D); выполнение тестов, Test Execution (E); отчетность о тестировании, Test Report (O). Показана роль оценки уровня развития СУТ в процессах управления развитием, предложен подход к определению оценки уровня развития СУТ. Подход основан на определении оценки уровня развития СУТ в условиях различных моделей разработки ПО, прежде всего с использованием экспертной оценки. В качестве методической основы использована классификация стадий зрелости проектного управления. Рассмотрены особенности и возможности применения как прямой, так и многокритериальной экспертной оценки. Предложено отображать вербальное описание стадий зрелости СУТ множеством частных количественных критериев, часть из которых может определяться инструментальными методами; для оценки значений других предлагается проводить многокритериальную экспертизу с участием узких специалистов по профилю каждого из критериев. Рассмотрены варианты алгоритмов одноуровневого определения глобального критерия уровня развития на основе множества оценок частных критериев: вычисление длины вектора в евклидовом пространстве и определение суммы взвешенных оценок частных критериев. Предложены двухуровневый вариант упорядочения частных критериев и соответствующие алгоритмы обработки множества их оценок, а также наглядная визуализация результатов оценки уровня развития СУТ. Разработанный на основе многокритериальной экспертной оценки подход позволяет определять степень зрелости СУТ ПО и целенаправленно управлять ее развитием.

Ключевые слова: *оценка, подход, тестирование, глобальный критерий, частный критерий, уровень развития, программные средства, многокритериальная оценка, экспертная оценка, система управления, бизнес-процесс.*

Разработка и производство *программных средств* (ПС) обрели все черты специальной отрасли, основу которой составляют соответствующие бизнес-процессы. От качества ПС существенно зависит эффективность выполнения бизнес-процессов во многих отраслях. Как следствие, в этой отрасли в полном объеме возникают и требуются решения задачи обеспечения и повышения эффективности или развития процессов и систем управления производственными бизнес-процессами. Управление бизнес-процессами становится все более сложным во всех сферах деятельности как по составу, так и функционально. Это обусловлено, прежде всего, усложнением выполняемых в составе бизнес-процесса задач, расширением их круга, повышением уровня требований к качеству их исполнения, причем число требований – критериев качества – постоянно растет. В основу современных систем управления положены *информационные технологии* (ИТ). Однако даже высокоэффективные ИТ не гарантируют успех управления, если бизнес-процессы и системы управления в целом как системы, обеспечивающие информационный менеджмент, не обладают уровнем развития,

достаточным для того, чтобы эффективно применять высокотехнологичные средства и процессы *обработки информации* (ОИ) [1–4].

Вследствие этого в отношении уровня развития систем управления может быть поставлена типовая задача управления, то есть задача целенаправленного формирования в системах управления заданного, или целевого, уровня развития: задание целевого уровня как стратегический план развития, оценка фактического уровня, сравнение его с целевым и внесение изменений в информационный менеджмент, то есть в систему управления. При этом должны быть формально поставлены и последовательно решены следующие задачи: определить понятие «уровень развития»; установить набор характеристик систем управления, определяющих ее уровень развития; описать связь значений характеристик системы с показателем уровня развития; сформировать воздействия на характеристики систем управления в нужном направлении. Специфика решения этих задач определяется условиями соответствующей сферы деятельности [2–4]. В настоящей статье эти вопросы рассматриваются применительно к особенностям управления разви-

тием системы управления тестированием при разработке ПС [5].

Тестирование ПС – это, с одной стороны, этап разработки ПО; с другой – специализированный бизнес-процесс в совокупности бизнес-процессов этой области деятельности. Понятие «тестирование ПС» определяется по-разному, однако бизнес-процессы тестирования по сути являются многофункциональными, а задачи управления развитием систем управления тестированием многокритериальными.

Как и в любой другой ИТ, в бизнес-процессе тестирования важным является экономический аспект, или эффективность ИТ: чем выше планируемые показатели качества тестирования, тем больше будут их трудоемкость и стоимость [6, 7]. Чем больше трудозатрат вкладывается в процесс тестирования, тем меньше ошибок в продукте остаются незамеченными. Однако совершенство в индустриальном программировании имеет пределы, которые прежде всего связаны с доступными затратами на получение программного продукта. В связи с этим определение трудоемкости бизнес-процесса тестирования конкретного продукта является оптимизационной задачей.

Стремление к уменьшению дефектов, или к повышению качества продукта, приводит к необходимости применения различных методов отладки и тестирования в процессе создания продукта. По мере обнаружения сложных ошибок и дефектов эффективность низкозатратных методов падает вместе с количеством обнаруживаемых ошибок. Отсюда следует, что каждый из методов тестирования имеет свою нишу, где он хорошо обнаруживает ошибки; вне этой ниши его эффективность падает. Поэтому необходимо совмещать различные методы и стратегии тестирования с целью обеспечения запланированного качества программного продукта при ограниченных затратах; это достигается при формировании адекватной системы управления тестированием.

Тестирование не является изолированным процессом, оно связано с другими работами по созданию ПС. Различные модели разработки ПС требуют различных подходов к тестированию.

Тестирование в каскадной модели. Каскадный подход хорошо зарекомендовал себя при построении ИС, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования, чтобы предоставить разработчикам свободу как можно лучшей с технической точки зрения их реализации.

Последовательная модель разработки (V-модель). Основная особенность V-модели – планомерное обеспечение тестирования на ранних стадиях проекта. V-модель является разновидностью каскадной модели: она также имеет последовательную структуру разработки, каждый новый шаг в ней начинается после завершения предыдущего.

Основная особенность V-модели заключается в том, что детализация проекта с течением времени возрастает при движении слева направо, и ни то, ни другое нельзя повернуть вспять. Итерации в проекте производятся по горизонтали, между левой и правой сторонами буквы. В модели в соответствии с введенными основными процессами выделены четыре уровня тестирования: компонентное (модульное), интеграционное, системное, приемочное.

Тестирование в спиральной модели. Спиральная модель – это модель, сочетающая в себе проектирование и поэтапное прототипирование. В процессе создания ПС для каждого витка спирали заново проводится исследование, ставятся цели, определяются конкретные характеристики проекта и составляется план проведения работ для следующего витка.

Процессная модель тестирования. Основной процесс тестирования состоит из следующих направлений деятельности:

- планирование и управление;
- анализ и проектирование;
- внедрение и реализация;
- оценка критериев выхода и создание отчетов;
- действия по завершению тестов.

Несмотря на логическую последовательность, действия в процессе могут накладываться друг на друга или происходить одновременно. Для конкретных системы и проекта обычно требуется адаптация этих направлений деятельности.

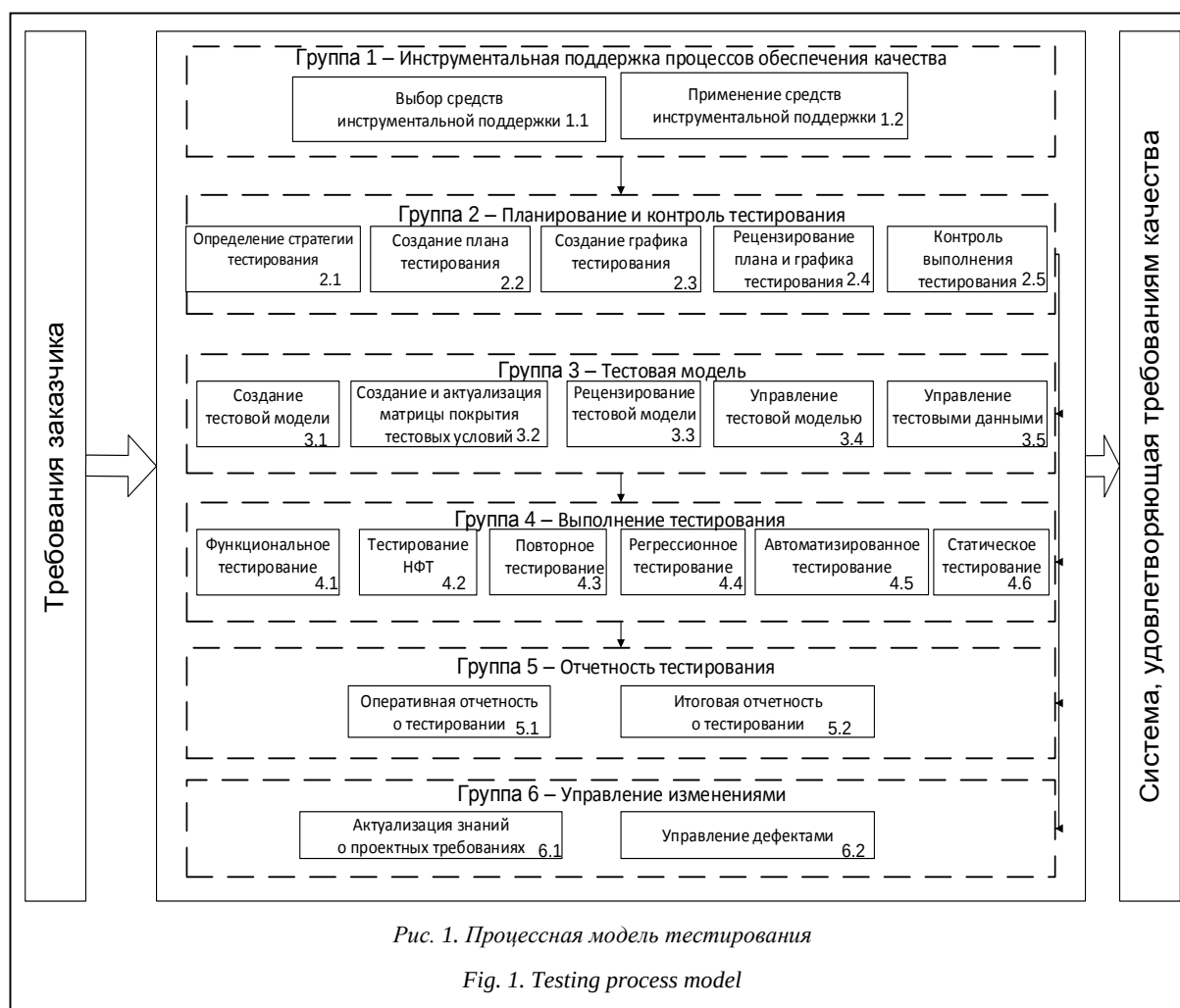
Функции процессной модели:

- определение, анализ, реинжиниринг и документирование производственных и организационных процессов отдела обеспечения качества;
- стандартизация процессов, ролей, артефактов и шаблонов, используемых в отделе;
- ускорение производственных процессов при сохранении высокого уровня качества;
- оптимизация производственных и организационных активностей, направленная на повышение их эффективности и сокращение трудозатрат отдела обеспечения качества.

Можно выделить шесть основных *групп процессов тестирования* (ГПТ), внутри которых можно произвести декомпозицию и выявить подпроцессы в рамках каждой ГПТ; рисунок 1 иллюстрирует декомпозицию ГПТ до подпроцессов.

Подход к оценке уровня развития системы управления тестированием

Тестирование – это подмножество, или поднабор, множества бизнес-процессов разработки ПО. Рассматривая тестирование как обособленный бизнес-процесс, в составе системы управления разработкой ПО в качестве подсистемы можно выделить систему управления тестированием, совершенствование которой обеспечивает повышение



качества выполнения бизнес-процесса тестирования. В задаче управления развитием системы управления тестированием, то есть в задаче целенаправленного формирования в системе управления тестированием заданного, или целевого, уровня развития, важное место занимает оценка фактического значения уровня развития. В качестве методической основы такой оценки в литературе достаточно широко используются классификации уровня развития или стадий зрелости (Maturity Model, MM – модель зрелости, англ.) различных систем и процессов [1, 6].

Известные модели оценки зрелости процессов разработки ПО – Capability Maturity Model (CMM), Capability Maturity Model Integration (CMMI), Test Maturity Model (TMM) – относятся к оценке процессов разработки ПО в целом, не выделяя и не оценивая отдельно систему управления тестированием. В настоящей работе в качестве методической основы оценки уровня развития, или стадии зрелости, систем управления тестированием предпринята попытка применения в условиях тестирования ПС классификации уровня развития проектного менеджмента IT Portfolio Management Maturity Model по М. Джеффри, которая содержит описание

четырёх уровней развития по мере их повышения: случайный, определенный, управляемый, согласованный [6].

Данная модель носит вербальный характер и основана на качественных оценках характеристик процессов и систем управления, оценивая которые, можно сделать заключение о том, на какой стадии находится рассматриваемая система. Ввиду качественного характера вербальной модели в этих условиях естественным является использование экспертной оценки.

Вербальную классификацию стадий зрелости как множество уровней развития можно представить в виде формальной модели – множества MM , каждому из элементов $S_i^M \mid i = \overline{1, 4}$ которого ставится в соответствие некий глобальный критерий G^M ; его значение определяет уровень развития; по существу это отображение множества классов на пространство значений глобального критерия:

$$MM = \{S_i^M \mid i = \overline{1, 4}\} \xrightarrow{f^G} G^M \in \overline{1, 4}. \quad (1)$$

Операция отображения f^G может быть реализована в виде *прямой экспертной оценки* (ПЭО), когда эксперт прямо называет номер класса в качестве оценки [6]. В сложных условиях, характерных

для такой экспертизы, обычно достаточно трудно найти компетентного эксперта, оценка которого вызывает безоговорочное доверие.

Степень доверия к оценке повышается при проведении коллективной экспертизы. Однако при участии в экспертизе нескольких экспертов оценки могут относиться к разным классам, из-за чего при обработке результатов опроса экспертов будут получены дробные значения классов. Поэтому значение глобального критерия, получаемое в результате экспертизы, будет принадлежать интервалу $\overline{0,4}$, то есть будет иметь вид числа $G_{МЭО}^M \in \overline{0,4}$. Таким образом, окончательное определение уровня развития – отображение значения глобального критерия на множество классов

$$G_{МЭО}^M \in \overline{0,4} \xrightarrow{f^S} S_i^M, \quad i = \overline{1,4}, \quad (2)$$

необязательно может давать целочисленную оценку, могут использоваться и дробные числа; при этом шкала классов служит как бы «каркасом» модели при реализации отображения f^S .

Однако на практике увеличить число экспертов сложно, поскольку безоговорочно авторитетных пока мало. К тому же такие эксперты требуют высокой оплаты; для многих компаний, прежде всего в сфере малого и среднего бизнеса, проведение экспертизы будет неподъемной затратой. Для решения этой проблемы предлагается метод *многокритериальной экспертной оценки* (МЭО): на основе вербального описания стадии зрелости вводятся количественные частные критерии P^M , отражающие развитие того или иного качества системы; значение каждого из критериев оценивается соответствующим профессиональным экспертом или измеряется; на основании их оценок уже расчетным путем определяется некий глобальный критерий – МЭО стадии зрелости $G_{МЭО}^M$, значение которой показывает уровень развития [6]. При этом вербальная классификация MM отображается множеством частных критериев P^M :

$$MM = \{S_i^M \mid i = \overline{1,4}\} \xrightarrow{f^P} P^M = \{P_j^M, D_j^P \mid j = \overline{1,N}\}, \quad (3)$$

где f^P – функция отображения; N – число введенных частных критериев; D_j^P – множество атрибутов частного критерия (наименование, описание и т.д.).

Экспертиза осуществляется в два этапа. На первом формируется состав множества частных критериев, его полнота и адекватность в отношении вербального описания определяются компетентностью состава привлеченных экспертов. Для повышения единообразия множество критериев целесообразно упорядочить следующим образом:

– все критерии нормировать и привести в диапазон значений $[0, 1]$: 0 – соответствующее свойство не проявляется, 1 – свойство проявляется в максимальной степени;

– критерии, отражающие в вербальной классификации убывающие свойства (недостатки), инвертировать и обратить в противоположные растущие свойства (достоинства).

На втором этапе экспертизы интервал значений $[0, 1]$ для каждого из критериев P_j^M представляется в виде строки, состоящей из 4 субинтервалов по числу уровней развития; при этом границы субинтервалов определяются экспертами по каждому из критериев. Следует отметить, что границы субинтервалов для разных критериев могут быть различными, то есть разбиение интервала $[0, 1]$ в общем случае неравномерно. Таким образом, состав множества критериев как структурная основа модели уровня развития принимает вид таблицы 1.

Таблица 1

Состав множества критериев

Table 1

The structure of the set of criteria

Критерий	Уровень развития			
	1	2	3	4
P_1^M				
.....				
P_N^M				

Оценка уровня развития формируется отображением множества частных критериев на значение глобального критерия:

$$P^M = \{P_j^M, D_j^P \mid j = \overline{1,N}\} \xrightarrow{f^{PG}} G_{МЭО}^M, \quad (4)$$

где операция отображения f^{PG} осуществляется на основе того или иного расчетного алгоритма; при этом могут использоваться различные подходы и методы. Далее для определения оценки уровня развития полученное значение $G_{МЭО}^M$ отображается на множество уровней развития $S_i^M \mid i = \overline{1,4}$ по шкале значений $G_{МЭО}^M$:

$$G_{МЭО}^M \xrightarrow{f^S} S_i^M \mid i = \overline{1,4}, \quad (5)$$

для чего должна быть построена шкала соответствия значений $G_{МЭО}^M$ и номеров уровней развития. Иначе говоря, для применения (4) и (5) необходимо разбить интервал возможных значений $G_{МЭО}^M$ на 4 субинтервала и присвоить им номера уровней развития от 1 до 4. Такая шкала строится на основе граничных значений частных критериев с использованием алгоритма, положенного в основу операции отображения f^{PG} . При этом будет осуществлен переход от вербального описания системы управления тестированием к формированию множества критериев системы и получению числового показателя стадии зрелости.

В условиях малого проектного предприятия применительно к специфике одного из проектов ПО сформировано множество $N=12$ частных критериев; для всех критериев определены шкалы из 4 субинтервалов (см. табл. 2).

Таблица 2

Структурная основа модели уровня развития

Table 2

A common framework of a development level model

Критерий	Вес	Значение критерия по стадиям зрелости			
		1	2	3	4
Группа 1. Проектные характеристики					
Планирование тестирования	0,2	0	[0;0,7]	[0,7; 0,8]	[0,8; 1]
Бюджет тестирования	0,3	[0; 0,3]	[0,3; 0,7]	[0,7; 0,8]	[0,8; 1]
Финансовые параметры	0,4	0	[0; 0,7]	[0,7; 0,9]	[0,9; 1]
Отчетность	0,1	0	[0; 0,5]	[0,5; 0,7]	[0,7; 1]
Границы субинтервалов ОГК по группе 1 – Cr_1^M		[0; 0,09]	[0,7; 0,68]	[0,68; 0,83]	[0,83; 1]
Группа 2. Организационные характеристики					
Роль процесса обеспечения качества в проекте	0,2	0	[0; 0,5]	[0,5; 0,7]	[0,7; 1]
Место ООК в организационной структуре предприятия	0,2	0	[0; 0,6]	[0,6; 0,7]	[0,7; 1]
Персонал ООК	0,4	[0; 0,5]	[0,5; 0,6]	[0,6; 0,8]	[0,8; 1]
Ценность данных и знаний для организации	0,2	0	[0; 0,5]	[0,5; 0,7]	[0,7; 1]
Границы субинтервалов ОГК по группе 2 – Cr_2^M		[0; 0,20]	[0,20; 0,56]	[0,56; 0,74]	[0,74; 1]
Группа 3. Общие характеристики					
Обмен информацией	0,2	[0; 0,4]	[0,4; 0,5]	[0,5; 0,7]	[0,7; 1]
Анализ деятельности	0,2	[0; 0,3]	[0,3; 0,4]	[0,4; 0,7]	[0,7; 1]
Производственные процессы	0,5	0	[0; 0,5]	[0,5; 0,8]	[0,8; 1]
Инновационная деятельность	0,1	0	0	[0; 0,5]	[0,5; 1]
Границы субинтервалов ОГК по группе 3 – Cr_3^M		[0; 0,14]	[0,14; 0,43]	[0,43; 0,73]	[0,73; 1]
Радиусы граничных сфер $G_{МЭОЗ}^M$		[0; 0,26]	[0,26; 0,98]	[0,98; 1,33]	[1,33; 1,73]

Для применения (4) и (5) на множестве полученных значений частных критериев, или для реализации операций отображения f^{PG} и f^S , необходимо задать вид этих операций, то есть сформировать алгоритм расчета количественной оценки уровня развития системы управления тестированием [6].

Так, в качестве глобального критерия может использоваться длина вектора в евклидовом пространстве, базисом которого являются значения частных критериев, то есть будет

$$G_{МЭОЗ}^M = \sqrt{\sum_{j=1}^{12} (P_j^M)^2}. \quad (6)$$

В качестве алгоритма определения глобального критерия может использоваться также взвешенное суммирование значений частных критериев:

$$G_{МЭОЗ}^M = \sum_{j=1}^{12} \phi_j \cdot P_j^M, \quad (7)$$

где ϕ_j , $j = 1, 12$ – весовой коэффициент, отражающий значимость j -го критерия в составе множества частных критериев, конкретные значения ϕ_j определяются экспертным путем; на сумму весовых коэффициентов налагается условие нормировки:

$$\sum_{j=1}^{12} \phi_j = 1. \quad (8)$$

Правда, наглядность таких глобальных критериев – вектора в 12-мерном пространстве и суммы взвешенных 12 слагаемых – невелика. В связи с этим множество частных критериев целесообразно представить в виде дерева, разбив его на группы по тем или иным признакам и сохраняя при этом наглядность как по числу критериев в группах, так и по числу групп. В этом случае алгоритм определения глобального критерия становится двухуровневым: сначала определяется в каждой группе *обобщенный групповой критерий* (ОГК), на основе множества которых далее определяется глобальный критерий. При этом на уровнях могут использоваться разные алгоритмы.

На основе исследования и анализа практики рассматриваемого проектного предприятия, процессной модели разработки ПС на предприятии, в частности, в сфере обеспечения качества программных продуктов в составе множества из 12 частных критериев, характеризующих деятельность отдела обеспечения качества, сформированы следующие группы характеристик – частных критериев по 4 критерия в каждой (см. табл. 2): проектные, организационные, общие.

Далее в условиях рассматриваемого предприятия в качестве алгоритма расчета ОГК использу-

ется взвешенное суммирование значений частных критериев, включенных в группу:

$$Gr_l^M = \sum_{k=1}^4 \alpha_{lk} \cdot P_{lk}^M, \quad l = \overline{1,3}, \quad (9)$$

где α_{lk} – весовые коэффициенты, отражающие значимость k -го критерия в составе множества частных критериев l -й группы, конкретные значения α_{lk} определяются экспертным путем; на сумму весовых коэффициентов в группе налагается условие нормировки:

$$\sum_{k=1}^4 \alpha_{lk} = 1, \quad l = \overline{1,3}. \quad (10)$$

В этих условиях значения всех обобщенных групповых критериев Gr_l^M , $l = \overline{1,3}$, попадают в интервал $[0, 1]$. Для каждого Gr_l^M , $l = \overline{1,3}$, на основе алгоритма (9) с учетом условия нормировки (12) интервал $[0, 1]$ разделен на 4 субинтервала, отражающих стадии зрелости в отношении данного ОГК (см. табл. 2).

Далее по аналогии с (5) выполняется отображение множества обобщенных групповых критериев на значение глобального критерия:

$$\{Gr_l^M, D_l^{Gr} \mid l = \overline{1,3}\} \xrightarrow{f^{GrG}} G_{МЭО}^M, \quad (11)$$

где D_l^{Gr} – множество атрибутов ОГК (наименование, описание и т.д.); f^{GrG} – функция отображения; при этом значение глобального критерия $G_{МЭО}^M$ определяется расчетным путем с использованием алгоритма, положенного в основу операции отображения f^{GrG} . Интервал возможных значений глобального критерия $G_{МЭО}^M$ разбивается на 4 субинтервала, отражающих стадии зрелости системы, с использованием алгоритма отображения f^{GrG} .

В условиях рассматриваемого предприятия в качестве глобального критерия используется длина вектора в евклидовом пространстве, базисом которого являются значения ОГК, то есть в данном случае глобальный критерий имеет вид

$$G_{МЭО}^M = \sqrt{\sum_{l=1}^3 (Gr_l^M)^2}. \quad (12)$$

С использованием этого алгоритма на основе граничных значений субинтервалов на шкалах ОГК интервал возможных значений глобального критерия $G_{МЭО}^M$ разбивается на 4 субинтервала. Таким образом, оценка уровня развития системы определяется по шкале соответствия субинтервалов значений глобального критерия $G_{МЭО}^M$ и шкалы уровней зрелости, то есть выполняется отображение значения $G_{МЭО}^M$ на шкалу классов:

$$G_{МЭО}^M \xrightarrow{f^S} S_i^M \mid i = \overline{1,4}, \quad (13)$$

где f^S – функция отображения, заключающаяся в применении правила соотнесения шкал значений $G_{МЭО}^M$ и номеров классов $S_i^M \mid i = \overline{1,4}$.

Точность экспертизы повышается при увеличении числа частных критериев N , что характеризует более детальное отражение вербального описания множеством количественно оцениваемых, пусть и экспертным путем, величин.

Здесь важно подчеркнуть, что для оценки того или иного частного критерия существенно проще найти компетентного эксперта, то есть узкого специалиста по данному критерию, оценка которого не подвергается сомнению и от которого не требуется столь широкая эрудиция, как от эксперта при проведении ПЭО.

В качестве визуальной иллюстрации модели оценки зрелости управления тестированием построена номограмма (рис. 2), на которой представлены области в пространстве глобальных критериев, отражающие все стадии зрелости.

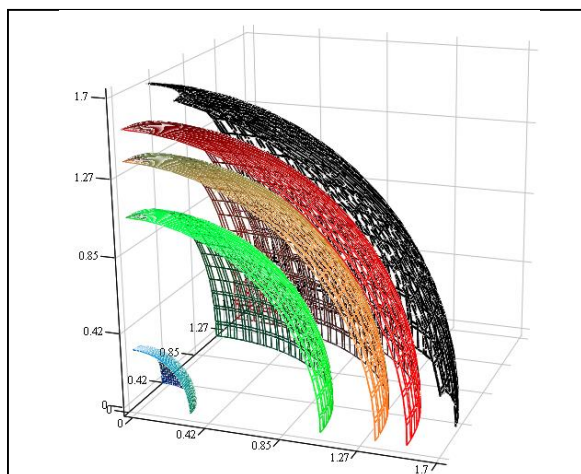


Рис. 2. Модель оценки уровня развития системы управления тестированием ПС

Fig. 2. An evaluation model of the software testing management system development

В порядке управления развитием систем управления тестированием полученное фактическое значение уровня развития сравнивается с его целевым значением. В случае, если целевой уровень не достигнут, необходимо выяснить, какие частные критерии вносят наибольший вклад в отставание в развитии, и провести мероприятия, обеспечивающие преодоление этого отставания. При этом важно оценить ресурсоемкость и стоимость соответствующих мероприятий, то есть эффективность процессов управления [7].

Таким образом, в статье поставлена задача формирования подхода к управлению уровнем развития системы управления тестированием в составе системы управления бизнес-процессами разработки ПС в условиях проектного предприятия. Подход основан на определении оценки уровня развития систем управления тестированием в условиях различных моделей разработки ПО, прежде всего с использованием многокритериальной экс-

пертой оценки. В качестве методической основы использована классификация стадий зрелости проектного управления. Предложено отображать вербальное описание стадий зрелости множеством частных количественных критериев, для оценки значений которых предлагается проводить многокритериальную экспертизу. Рассмотрены варианты алгоритмов по этапам определения глобального критерия уровня развития. Предложенный на основе многокритериальной экспертной оценки подход позволяет оценивать степень зрелости систем управления тестированием ПО и целенаправленно управлять ее развитием.

Литература

1. Жданович О.А., Корнюшко В.Ф., Иванчук И.С., Костров А.В. Степень готовности системы управления бизнес-про-

цессами к внедрению информационных технологий (методика оценки) // Прикладная информатика. № 2 (50). 2014. С. 14–22.

2. Мухин К.О., Костров А.В. Описание моделей базовых элементов объектно-ориентированной модели производственных процессов для нахождения оптимального управления // Научные технологии. 2013. Т. 14. № 4. С. 62–67.

3. Мухин К.О., Костров А.В. Метод применения объектно-ориентированных имитационных моделей для управления сложными производственными процессами // Нелинейный мир. 2013. Т. 11. № 5. С. 332–337.

4. Решетников В.Н. Космические телекоммуникации. Системы спутниковой связи и навигации. СПб, 2010. 134 с.

5. Якобсон А., Буч Г., Рамбо Дж. Унифицированный процесс разработки программного обеспечения; [пер. с англ.]. СПб: Питер, 2002. 496 с.

6. Костров А.В., Егорова И.В., Жданович О.А. Подход к управлению уровнем развития информационных систем // Динамика сложных систем. 2015. Т. 9. № 1. С. 24–31.

7. Костров А.В., Матвеев Д.А. Информационный менеджмент. Оценка эффективности информационных систем. Владимир: Изд-во ВлГУ, 2004. 116 с.

DOI: 10.15827/0236-235X.112.126-132

Received 06.07.15

AN APPROACH TO SOFTWARE TESTING MANAGEMENT SYSTEM DEVELOPMENT

Kornyushko V.F., Dr.Sc. (Engineering), Professor, Head of Chair
(Lomonosov Moscow State University of Fine Chemical Technologies,
Vernadsky Ave. 86, Moscow, 119571, Russian Federation);

Kostrov A.V., Dr.Sc. (Engineering), Professor, akostrov@rambler.ru;

Porodnikova P.A., Postgraduate Student, polina.porodnikova@mail.ru
(Alexander and Nikolay Stoletovs Vladimir State University,
Gorkogo St. 87, z. Vladimir, 600000, Russian Federation)

Abstract. The paper considers the problem of creating an approach to testing management system (TMS) development control as a part of software development business process management system in a project company. The authors suggest to separate software testing process as an independent business process, so software development management system has a testing management subsystem. The paper reviews the features of testing management in typical software development models. There are the variants of basic testing processes implementation: Review (R); Test Design (D); Test Execution (E); Test Report (O). The article shows the role of TMS development evaluation in the development management processes, and provides the approach to its evaluation. The approach is based on TMS development evaluation for different software development models, expert evaluation in particular. Classification of project management maturity stages is used as a methodological basis. The paper considers the features and applicability of both direct and multi-criteria expert evaluation. The authors suggest displaying a verbal description of TMS maturity stages as a set of private quantitative criteria; some of them can be determined by instrumental methods. To evaluate the others it is proposed to make multi-criteria assessment with the assistance of field experts according to each criterion. The paper considers the variants of single-level algorithms to determine a development level global criteria based on multiple assessments of partial criteria: calculation of the length of the vector in Euclidean space and determining the amount of the weighted assessments of partial criteria. The paper proposes a two-level ordering of the partial criteria and corresponding algorithms for processing their set of estimates, as well as a visualization of the results of the TMS development evaluation. The described approach allows assessing the software TMS maturity and managing its development.

Keywords: estimate, approach, testing, global criterion, partial criterion, development level, software, multi-criteria estimate, expert estimate, management system, business process.

References

1. Zhdanovich O.A., Kornyushko V.F., Ivanchuk I.S., Kostrov A.V. The estimate methodics of business process management system readiness level to information technology introduction. *Prikladnaya informatika* [Applied Informatics]. 2014, no. 2 (50), pp. 14–22 (in Russ.).

2. Mukhin K.O., Kostrov A.V. Model description of the basic elements of object-oriented model of the production process for determining optimal control. *Naukoemkie tekhnologii* [Science Intensive Technologies]. 2013, vol. 14, no. 4, pp. 62–67 (in Russ.).

3. Mukhin K.O., Kostrov A.V. The Method of a object-oriented simulation models Application for the complex production process Control. *Nelineyny mir* [Nonlinear World]. 2013, vol. 11, no. 5, pp. 332–337 (in Russ.).

4. Reshetnikov V.N. *Kosmicheskie telekommunikatsii. Sistemy sputnikovoy svyazi i navigatsii* [Space telecommunications. Satellite Communication Systems and Navigation]. St.-Petersburg, 2010, 134 p.

5. Booch G., Rumbaugh J., Jacobson I. *The Unified Modeling Language. User Guide*. Addison Wesley Publ., 1999, 482 p. (Russ. ed.: St.-Petersburg, Piter Publ., 2002, 496 p.).

6. Kostrov A.V., Egorova I.V., Zhdanovich O.A. The approach to information system development level control. *Dinamika slozhnykh sistem* [Dynamics of Complex Systems]. 2015, vol. 9, no. 1, pp. 24–31 (in Russ.).

7. Kostrov A.V., Matveev D.A. *Informatsionny menedzhment. Otsenka effektivnosti informatsionnykh sistem* [Information management. Information system performance evaluation]. Vladimir, VISU Publ., 2004, 116 p.

УДК 681.322

Дата подачи статьи: 08.05.15

DOI: 10.15827/0236-235X.112.133-141

ЗАДАЧИ ОБЕСПЕЧЕНИЯ УСТОЙЧИВОСТИ ФУНКЦИОНИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

*Д.О. Есиков, аспирант, mcgeen4@gmail.com
(Тульский государственный университет,
пр. Ленина, 92, г. Тула, 300012, Россия)*

Предложены способы обеспечения устойчивости функционирования распределенных информационных систем. Формализованы задачи обеспечения устойчивости функционирования распределенных информационных систем в составе: математическая модель оптимизации распределения элементов ПО функциональных задач по узлам сети, математическая модель оптимизации распределения информационных ресурсов по центрам хранения и обработки данных, математическая модель определения рационального уровня расходов на формирование комплекса средств хранения данных в центрах хранения и обработки информации, математическая модель оптимизации состава технических средств системы хранения и обработки данных, математическая модель оптимизации распределения резерва информационных ресурсов по центрам хранения и обработки данных. Показано, что они относятся к классу задач дискретной оптимизации, дана их характеристика. Предложен порядок применения комплекса математических моделей обеспечения устойчивости функционирования распределенных информационных систем с указанием входных и выходных данных для каждой модели. Для решения разработанных задач предложено использовать программный комплекс, реализующий метод ветвей и границ по решению задач дискретной оптимизации с булевыми переменными с применением алгоритма предварительного определения порядка ветвления переменных на основе использования теории двойственности. Применение теории двойственности в методе ветвей и границ позволяет существенно усилить отсев бесперспективных вариантов и сократить время решения задач по сравнению с традиционным методом в среднем в 8 раз. Приведены особенности практического использования разработанного комплекса математических моделей на стадиях проектирования, эксплуатации и совершенствования жизненного цикла распределенных информационных систем.

Ключевые слова: *распределенная информационная система, математические модели, дискретная оптимизация, оптимизация по Парето, устойчивость функционирования.*

Задача обеспечения устойчивости функционирования *распределенных информационных систем* (РИС), несмотря на интенсивное развитие средств хранения и обработки информации, до сих пор стоит очень остро. Данное обстоятельство связано с ростом объемов критически важных данных, хранящихся и обрабатываемых в системе, с ограниченными материальными средствами, выделяемыми на создание и развертывание РИС, ограниченной пропускной способностью каналов передачи данных и других характеристик сети, с влиянием дестабилизирующих факторов различной природы.

На предприятиях промышленности, в организациях и государственных учреждениях различных уровней эксплуатируется большое количество *информационных систем* (ИС), требующих хранения больших объемов данных, критичных для решения задач управления. К таким ИС относятся системы учета и хранения конструкторской документации, системы электронного документооборота, ИС управления персоналом, ERP-системы и др.

Надежное и бесперебойное функционирование данных систем является залогом эффективного функционирования организаций и предприятий в различных сферах экономики. Под устойчивостью функционирования системы понимается ее способность выполнять возложенные функции с заданными показателями качества в условиях воздействия дестабилизирующих факторов [1, 2].

В настоящее время наиболее эффективными для обеспечения устойчивости функционирования РИС являются следующие подходы [3, 4]:

- совершенствование информационно-вычислительного процесса, в том числе в части обеспечения его устойчивости;
- создание высоконадежных подсистем хранения и резервирования данных.

В общем случае можно утверждать, что устойчивость функционирования РИС непосредственно зависит от устойчивости процессов *хранения и обработки данных* (ХОД) в системе вычислительных средств распределенной сети. Для повышения устойчивости процессов ХОД на этапе проектирования эксплуатации РИС целесообразно рациональное распределение элементов ПО и информационных массивов по узлам вычислительной сети, а при воздействии дестабилизирующих факторов, в случае структурной деградации сети – их перераспределение [1].

Решение задач организации процессов ХОД позволяет повысить эффективность функционирования системы за счет снижения издержек, связанных с разрушением информации, включая потери от невозможности решения задач и получения недостоверных результатов, за счет уменьшения времени и затрат на восстановление разрушенной или искаженной информации, рационального использования внешней памяти вычислительных средств для размещения данных РИС и т.д.

Практическое решение вопросов оптимизации распределения (перераспределения) информационных ресурсов, а также управление данным распределением в РИС связаны с развитием теории дискретной оптимизации. Большая размерность, дискретность, нелинейность указанных задач и необходимость их решения в жестких временных рамках (на этапах эксплуатации и совершенствования) порождают задачу многомерности, дискретности и поиска путей снижения размерности сформулированных задач, разработки и использования эффективных методов их решения.

Основу РИС составляет система ХОД, предназначенная для организации надежного, а также отказоустойчивого хранения данных, высокопроизводительного доступа серверов к устройствам системы ХОД.

Для рациональной организации процессов ХОД в РИС в структуре системы ХОД создают центры ХОД, основной задачей которых является реализация эффективных, с точки зрения решаемых в системе задач, процессов хранения и обработки информации.

Вследствие ограниченности материальных средств, выделяемых на создание РИС, хранение и обработка информации в них осуществляются в ограниченном количестве центров ХОД, создаваемых в соответствующих узлах сети. В центрах ХОД аккумулируются мощные вычислительные средства и средства хранения данных, обеспечивающие решение функциональных задач как отдельного сегмента сети, так и системы в целом.

Таким образом, для обеспечения устойчивости функционирования РИС разработан комплекс математических моделей (рис. 1) в следующем составе:

- оптимизация распределения элементов ПО функциональных задач по узлам сети;
- оптимизация распределения информационных ресурсов по центрам ХОД;
- определение рационального уровня расходов на формирование комплекса средств хранения данных в центрах ХОД;
- оптимизация состава технических средств системы ХОД;
- оптимизация распределения резерва информационных ресурсов по центрам ХОД.

Задача определения числа и месторасположения центров ХОД в РИС решается с целью получения такого плана распределения центра ХОД, который обеспечивал бы использование наиболее скоростных каналов передачи данных при информационном обмене. Определение количества и месторасположения центров ХОД в системе ХОД возможно путем решения задачи о p -медиане [5], которая относится к классу задач целочисленного линейного программирования. Для ее решения обычно используется метод ветвей и границ.

Целью решения задачи оптимизации состава технических средств системы ХОД, составляющих основу центра ХОД, является получение такого состава технических средств, который обеспечивал бы максимальное значение коэффициента готовности системы в целом при соблюдении ограничений на общий объем пространства средств хранения, на время восстановления системы, на стоимость системы резервирования.

Пусть все средства хранения данных, претенденты на включение в состав комплекса технических средств центра ХОД, делятся на I групп. В каждую i -ю группу средств хранения входят Z_i средств хранения, имеющих номера с 1 по Z_i . Заданы стоимость, время восстановления и объем информационного пространства каждого средства хранения данных.

Тогда по критерию максимума значения коэффициента готовности системы хранения данных в целом задача оптимизации состава средств хранения может быть сформулирована следующим образом [6]. Найти такие значения переменных z_i ($i=1, 2, \dots, I$), что

$$\prod_{i=1}^I K_{r_i}(z_i) \rightarrow \max \quad (1)$$

при ограничениях

- на стоимость системы резервирования:

$$\sum_{i=1}^I C_i(z_i) \leq C^{\text{доп}}, \quad (2)$$

- на время восстановления системы резервирования:

$$\sum_{i=1}^I T_i^B(z_i) \leq T^B_{\text{доп}}, \quad (3)$$

- на общий объем средств хранения:

$$\sum_{i=1}^I V_i(z_i) \geq V_i^{\text{доп}}, \quad (4)$$

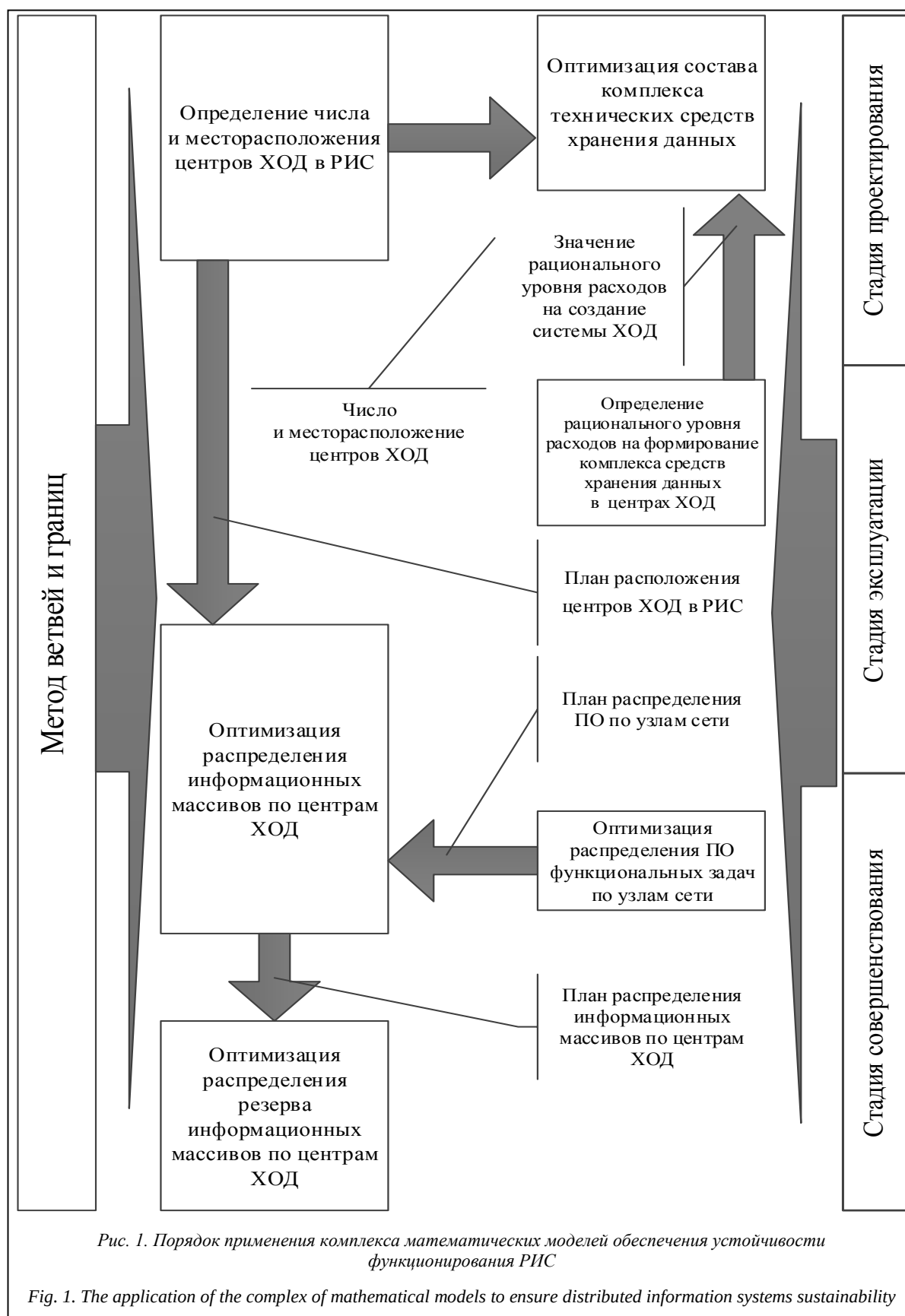
- на значения переменных:

$$z_i = 1, 2, \dots, Z_i, \quad (5)$$

где $C^{\text{доп}}$ – максимально допустимая стоимость системы хранения и резервирования; $T^B_{\text{доп}}$ – максимально допустимое время восстановления системы хранения и резервирования; $V_i^{\text{доп}}$ – максимально допустимый информационный объем системы хранения и резервирования данных; I – число видов средств хранения информации.

Задача (1)–(5) относится к классу задач выбора комплекса технических средств, и для ее решения применяются методы динамического программирования [7].

В связи с тем, что затраты на построение системы ХОД в центре ХОД существенны по размеру и значительно растут при ужесточении требований к надежности показателям качества их функционирования, необходимо обоснование рациональности выбора их размера.



Для определения рационального уровня расходов на формирование комплекса средств хранения

данных в центрах ХОД предлагается использовать методику, основанную на применении элементов

теории вероятностей и теории благосостояния (принцип оптимальности по Парето) [8, 9], суть которой состоит в следующем.

Пусть q – параметр, характеризующий уровень сохранности информации в РИС, $0 \leq q \leq 1$. Обозначим через Z_Φ потери от разрушения или нарушения целостности информации в РИС, которые зависят от определенного в РИС уровня *обеспечения сохранности информации* (ОСИ), описываемого значением параметра q . Будем считать, что эта величина принимает максимальное значение при $q = 0$ (отсутствие системы ОСИ) и значение, равное нулю, при $q=1$ (наличие «абсолютной» системы ОСИ). Таким образом, имеем функцию вида $Z_\Phi = Z_\Phi(q)$, $q \in [0, 1]$.

При отсутствии системы ОСИ ($q = 0$) потери от разрушения или искажения информации в РИС составляют C_1 единиц. Будем считать, что с ростом величины q значение потерь Z_Φ уменьшается по экспоненциальному закону и при $q=1$ становится практически равным нулю. Этим требованиям удовлетворяет функция $Z_\Phi(q) = C_1 e^{-\alpha q}$.

Параметр α , входящий в это выражение, можно вычислить из уравнения $\alpha = \ln\left(\frac{C_1}{\varepsilon}\right)$, где ε – параметр, учитывающий степень важности информации, обращающейся в РИС.

Пусть Z_c – затраты на создание системы ОСИ, которые описываются функцией вида $Z_c = Z_c(q)$, $q \in [0, 1]$.

Эта функция описывается зависимостью $Z_c(q) = C_2 e^{-b(1-q)}$.

Здесь C_2 – стоимость создания «абсолютной» системы ОСИ, при которой уровень сохранности информации $q = 1$. Параметр b определяется по формуле $b = \ln\left(\frac{C_2}{\varepsilon}\right)$.

При разработке системы ОСИ естественным требованием является минимизация затрат Z_Φ и Z_c путем выбора значения вероятности q на интервале $[0, 1]$, то есть $Z_\Phi(q) \rightarrow \min_{0 < q < 1}$, $Z_c(q) \rightarrow \min_{0 < q < 1}$.

Данные целевые функции противоречивы, так как с ростом значения q затраты Z_Φ убывают, а затраты Z_c возрастают. Это позволяет установить некоторый компромисс между значениями затрат Z_Φ и Z_c .

Для реализации такого подхода сформулируем двухкритериальную задачу оптимизации вида $(Z_\Phi, Z_c) \rightarrow \min_{0 < q < 1}$.

Парето-оптимальное решение этой задачи будем искать путем минимизации линейной свертки критериев: $L(q, \lambda) = \lambda Z_\Phi(q) + (1 - \lambda) Z_c(q) \rightarrow \min_{0 < q < 1}$.

Здесь $\lambda \in (0, 1)$ – параметр свертки критериев.

Для решения задачи можно использовать необходимое условие экстремума функции $L(q, \lambda)$, которое записывается как

$$\frac{\partial L}{\partial q} = \lambda Z'_\Phi(q) + (1 - \lambda) Z'_c(q) = 0.$$

Решая это уравнение, получаем параметрическую зависимость вида $q = q(\lambda)$, $\lambda \in (0, 1)$, которая описывает множество парето-оптимальных решений задачи в пространстве решений.

После подстановки функций $Z_\Phi = Z_\Phi(q)$ и $Z_c = Z_c(q)$ в условие экстремума $L(q, \lambda)$ получим [7]:

$$-\lambda \alpha C_1 e^{-\alpha q} + (1 - \lambda) C_2 b e^{-b(1-q)} = 0.$$

Решая это уравнение, получим конкретный вид зависимости:

$$q(\lambda) = \frac{-\ln\left(\frac{C_2 b (1 - \lambda)}{\lambda C_1 \alpha}\right) + b}{\alpha + b}, \quad \lambda \in (0, 1).$$

Для определения допустимых значений параметра свертки λ , входящего в это выражение, примем, что задано значение q^* , определяющее границу вероятности q , то есть минимально допустимый с точки зрения ОСИ уровень. Тогда парето-оптимальные значения вероятности $q(\lambda)$ должны удовлетворять неравенству вида $q^* \leq q(\lambda) \leq 1$.

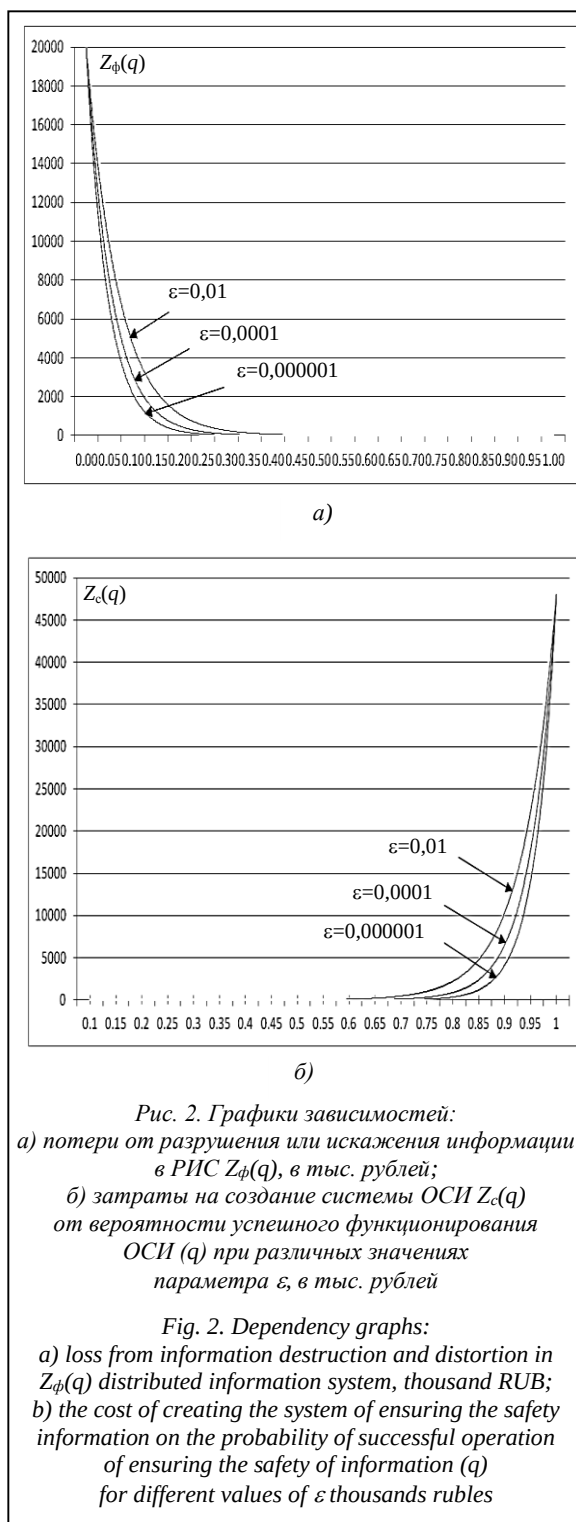
При этом минимальное значение $\lambda_{\min} \in (0, 1)$ определяется как $\lambda_{\min} = \frac{C_2 b e^{-b(1-q^*)}}{C_2 b e^{-b(1-q^*)} + C_1 \alpha e^{-\alpha q^*}}$.

При $q(\lambda) < 1$ имеем следующее ограничение на значение параметра λ : $\lambda_{\max} = \frac{C_2 b}{C_2 b + C_1 \alpha e^{-\alpha q^*}}$ при замене q^* на величину $q(\lambda) = 0,85$.

В связи с тем, что уровень ОСИ, при котором $q=1$, на практике недостижим, но разработчик стремится обеспечить такой уровень, значение C_2 можно формировать двумя путями: заданием разработчиком величины C_2 , при которой, с его точки зрения, будет достигнут максимально возможный уровень ОСИ в РИС (30–50 % стоимости системы ОСИ), и использованием значения C_2 из практики разработки и функционирования хорошо защищенных систем ХОД.

Так, при $q^*=0,85$, $C_1=150\,000\,000$, $C_2=48\,000\,000$ можно получить множество значений уровня обеспечения ОСИ путем варьирования значений параметра λ в интервале $[0,999977, 0,999999]$.

Пример графиков функций $Z_\Phi(q)$ и $Z_c(q)$ при $\varepsilon = 0,01$ для особо важной информации (ИС учета и хранения конструкторской документации), $\varepsilon = 0,0001$ для важной информации (информационных массивов электронного документооборота, договорные материалы), $\varepsilon = 0,000001$ для информационных массивов, хранящих информацию, важную в течение заданного периода времени (текущая переписка), представлен на рисунке 2.



Из рисунка 2 видно следующее:

- график зависимости $Z_{c1}(q)$ является некорректным, поскольку при затратах на разработку систем информационной безопасности, равных 0, вероятность обеспечения безопасности при функционировании системы ХОД $q \rightarrow 0$;

- при уменьшении величины ε наблюдается уменьшение значения функций $Z_\phi(q)$ и $Z_c(q)$ при одинаковом значении аргумента.

Исходя из этого, можно сделать вывод о том, что для систем ОСИ, в которых обращается информация с более высокой степенью важности, минимальное значение в интервале варьирования параметра λ должно быть больше, чем для системы ХОД с более низкой степенью важности.

Таким образом, множество значений вероятности обеспечения ОСИ в РИС и соответствующие им значения потерь от потери или искажения информации и затрат на создание системы ОСИ возможно определить путем варьирования значений параметра свертки λ в интервале $[\lambda_{\min}, \lambda_{\max}]$, что позволит выбрать значение требуемой вероятности ОСИ из полученного множества. При этом в качестве исходных данных для получения вышеуказанного множества выступает незначительный набор данных.

На основе результатов решения задач определения числа и месторасположения центров ХОД в РИС, оптимизации состава технических средств системы ХОД формализованы задачи оптимизации распределения элементов ПО функциональных задач по узлам сети и оптимизации распределения резерва информационных ресурсов по центрам ХОД.

Для простоты дальнейших рассуждений примем следующее допущение: время получения информации из информационного массива равно 0, то есть оперативность получения данных для решения задач не зависит от каналов передачи данных и размещения информационного массива по узлам сети.

Пусть задана сеть, состоящая из L вычислительных комплексов.

В сети решается K задач. Для каждого j -го узла сети сформирован и определен перечень задач, решение которых разрешено должностным лицам. Решение каждой из задач связано с формированием и обработкой соответствующего числа запросов на получение (сохранение, изменение) данных из (в) информационных массивов.

Размещение ЭПО по узлам сети определяется планом распределения, задаваемым матрицей

$$\Phi = \|\varphi_{kj}\|,$$

$$\varphi_{kj} = \begin{cases} 1, & \text{если } k - \text{й ЭПО размещается} \\ & \text{на } j - \text{м узле сети,} \\ 0 & \text{в противном случае.} \end{cases}$$

Тогда выражения для определения вероятности успешного решения P_{jk} , времени решения задач $T_{jk}^{\text{реш}}$, объема информации, передаваемой по каналам связи при решении каждой задачи на каждом вычислительном комплексе (узле сети) $\Lambda_{jk}(j=1, 2, \dots, L; k=1, 2, \dots, K)$, имеют вид

$$P_{jk} = \tau_{jk} \sum_{l=1}^L \varphi_{kl} P_{jkl}^n,$$

$$T_{jk}^{\text{реш}} = \sum_{l=1}^L \varphi_{kl} (T_{jkl} + t_{jkl}^{\text{реш}}),$$

$$\Lambda_{jk} = \lambda_{jk} \left[\sum_{i=1}^L \left((F_{ji} I_{jk}^3 + F_{ij} I_{jk}^0) \varphi_{ki} \right) \right],$$

где P_{jk}^n – вероятность успешной передачи информации между узлами j и i при решении на j -м вычислительном комплексе k -й задачи, $P_{jk}^n = P_{ijk}^3 P_{ijk}^c$; P_{ijk}^3, P_{ijk}^c – вероятность успешной передачи запроса (сообщения) на решение и сообщения с результатом решения должностным лицом j -го узла сети k -й задачи в i -м узле сети соответственно; F_{ij} – число ИС, пересылаемых при передаче из i -го узла сети в j -й; T_{ijk} – среднее время передачи сообщения из i -го узла сети в j -й при решении k -й задачи должностным лицом j -го вычислительного комплекса: $T_{ijk} = T_{ijk}^3 + T_{ijk}^c$, T_{ijk}^3, T_{ijk}^c – среднее время передачи запроса на решение и сообщения, содержащего результаты решения должностным лицом j -го вычислительного комплекса k -й задачи в i -м узле сети соответственно; $t_{jkl}^{\text{реш}}$ – время выполнения k -го ЭПО на l -м вычислительном комплексе должностным лицом j -го узла при наличии всех исходных данных; $\tau_{jk} = 1$, если должностные лица j -го вычислительного комплекса решают k -ю задачу, $\tau_{jk} = 0$ в противном случае; λ_{jk} – интенсивность решения k -й задачи должностными лицами j -го вычислительного комплекса; I_{jk}^3 – объем запроса, формируемого должностными лицами j -го вычислительного комплекса на решение k -й задачи; I_{jk}^c – объем сообщения с результатами решения должностными лицами j -го вычислительного комплекса k -й задачи; u_k – объем внешней памяти, занимаемый k -м ЭПО.

С использованием данных выражений задача распределения ЭПО по критерию минимума объема информации, передаваемой по каналам связи при выполнении в РИС всех ЭПО (при решении всех функциональных задач), может быть сформулирована следующим образом. Определить такие значения φ_{kj} ($k=1, 2, \dots, K; j=1, 2, \dots, L$), что

$$\Lambda = \min \sum_{j=1}^L \sum_{k=1}^K \Lambda_{jk}(\varphi) \quad (6)$$

при ограничениях

– на время решения задач:

$$T_{jk}^{\text{реш}} \leq T_{jk}^{\text{доп}}, j=1, 2, \dots, L, k=1, 2, \dots, K; \quad (7)$$

– на вероятность успешного решения задач:

$$P_{jk} \geq P_{jk}^{\text{доп}}, j=1, 2, \dots, L, k=1, 2, \dots, K; \quad (8)$$

– на объем исходного информационного пространства:

$$\sum_{k=1}^K \varphi_{kj} u_k \leq V_j, j=1, 2, \dots, L; \quad (9)$$

– на значения переменных:

$$\sum_{j=1}^L \varphi_{kj} = 1, k=1, 2, \dots, K; \quad (10)$$

$$\varphi_{kj} = \{0, 1\}; k=1, 2, \dots, K; j=1, 2, \dots, L, \quad (11)$$

где $T_{jk}^{\text{доп}}$ – допустимое значение времени решения k -й задачи должностными лицами j -го вычислительного комплекса; $P_{jk}^{\text{доп}}$ – допустимое значение вероятности успешного решения k -й задачи должностными лицами j -го вычислительного комплекса; V_j – максимально допустимый объем информационного пространства.

Задача (6)–(11) относится к классу линейных задач дискретного программирования с булевыми переменными и со смешанными ограничениями.

План распределения программ по узлам сети используется в качестве исходных данных для задачи распределения информационных массивов по центрам ХОД. В качестве информационного массива могут рассматриваться БД, массив электронных образов документов, обрабатываемых и циркулирующих в системе, и т.п. Необходимо распределить информационные массивы по центрам ХОД таким образом, чтобы минимизировать время решения всех задач в РИС.

Для простоты дальнейших выводов используем следующие допущения:

– информационный обмен между ЭПО при решении задач осуществляется посредством сохранения промежуточных и конечных результатов в информационные массивы;

– время выполнения запросов к информационным массивам пренебрежительно мало по сравнению со временем передачи информации по каналам связи.

Введем следующие обозначения: M – количество информационных массивов; $\|\varphi_{lj}\|$ – матрица распределения ЭПО по узлам сети ($l=1, 2, \dots, K, j=1, 2, \dots, L$), $\varphi_{lj} = \{0, 1\}$; $\|\omega_{lm}\|$ – матрица связи ЭПО и информационных массивов ($l=1, 2, \dots, K, m=1, 2, \dots, M$), $\omega_{lm} = 1$, если l -й ЭПО имеет информационный обмен в процессе своего решения с m -м информационным массивом, и 0 в противном случае; $\|\psi_{mj}\|$ – матрица распределения информационных массивов ($m=1, 2, \dots, M, j \in X_p$); L_{lm} – среднее значение суммарного объема информационного обмена l -го ЭПО при его выполнении с m -м информационным массивом; l_0 – длина эталонного сообщения; t_{ij}^0 – среднее время доведения эталонного сообщения от i -го узла сети к j -му; t_{ij}^p – среднее время выполнения l -го ЭПО в j -м узле без учета информационного обмена с информационными массивами; X_p – множество номеров узлов сети, в которых создаются центры ХОД (результат решения задачи о p -медиане).

Тогда задача оптимизации распределения информационных массивов по критерию минимума среднего времени решения всех задач в РИС может быть сформулирована следующим образом.

Определить такие значения ψ_{mj} , $m=1, 2, \dots, M$, $j \in X_p$, что $T = \min \sum_{j=1}^L \sum_{k=1}^K \varphi_{kj} \left(t_{kj}^p + \sum_{l=1}^M \varpi_{kl} \frac{L_{kl}}{l_j} \sum_{m \in X_p} \psi_{lm} t_{jm}^p \right)$

при ограничениях

– на объем пространства на носителях для хранения информационных массивов:

$$\sum_{k=1}^M \psi_{kj} u_k \leq V_j, \quad j \in X_p;$$

– на значения переменных:

$$\sum_{j=1}^L \psi_{kj} = 1, \quad \forall k, \quad \psi_{kj} = \{0, 1\}.$$

Данная задача относится к классу задач целочисленного линейного программирования с булевыми переменными [10, 11]. Для ее решения может быть применен метод ветвей и границ.

Задача оптимизации распределения резерва информационных ресурсов по центрам ХОД решается с целью получения такого плана распределения резерва информационных ресурсов по центрам, который обеспечивал бы минимальное время восстановления работоспособности системы ХОД в случае нарушения целостности информации при соблюдении ограничений на стоимость хранения резерва информационных ресурсов.

Пусть N – число центров ХОД, создаваемых в системе; M – число информационных массивов, хранящихся и обрабатываемых в центрах ХОД. Известны распределение информационных массивов по центрам ХОД, задаваемое матрицей $\|x_{ij}\|$, $i=1, 2, \dots, M$, $j=1, 2, \dots, N$, а также объемы каждого i -го информационного массива V_i и пропускная способность каналов передачи данных между каждым l -м и m -м центром ХОД – v_{lm} . Тогда можно определить значение среднего времени восстановления i -го информационного массива в случае его искажения или разрушения: $T_i^B = \sum_{l=1}^N x_{il} \sum_{m=1}^N y_{im} \left(\frac{V_i}{v_{lm}} + t_{il}^p \right)$,

где t_{il}^p – время выполнения работ по развертыванию и подключению (регламентных работ) i -го информационного массива после его восстановления в l -м центре ХОД; $\|y_{ij}\|$, $i=1, 2, \dots, M$, $j=1, 2, \dots, N$ – матрица распределения резерва информационного массива по центрам ХОД, причем $y_{ij} = \{0, 1\}$, $\sum_{j=1}^N y_{ij} = 1$, $i=1, 2, \dots, M$, $x_{ij}, y_{ij}=0 \quad \forall i, j$.

В этом случае задача распределения резерва информационного массива по центрам ХОД по критерию минимума общего значения среднего времени восстановления информации в системе ХОД [12, 13] может быть сформулирована следующим образом.

Определить такие значения $\|y_{ij}\|$, $i=1, 2, \dots, M$, $j=1, 2, \dots, N$, что

$$\sum_{i=1}^M \sum_{l=1}^N x_{il} \sum_{m=1}^N y_{im} \left(\frac{V_i}{v_{lm}} + t_{il}^p \right) \rightarrow \min \quad (12)$$

при ограничениях

– на стоимость хранения резерва информационного массива:

$$\sum_{j=1}^N C_j \sum_{i=1}^M y_{ij} \left[\frac{V_i}{v} \right] \leq C_{\text{доп}}; \quad (13)$$

– на значения переменных:

$$y_{ij} = \{0, 1\}, \quad (14)$$

$$\sum_{j=1}^N y_{ij} = 1, \quad i=1, 2, \dots, M, \quad x_{ij}, y_{ij}=0 \quad \forall i, j, \quad (15)$$

где C_j – стоимость хранения единицы объема информации в j -м центре ХОД; v – размер единицы объема хранения информации; $C_{\text{доп}}$ – максимально возможное значение стоимости хранения резерва информационного массива в системе ХОД.

Задача (12)–(15) относится к классу задач дискретного линейного программирования с булевыми переменными и может быть сведена к задаче о ранце [14], для решения которой может использоваться метод ветвей и границ [14]. Ее решение позволит минимизировать время восстановления работоспособности системы ХОД в случае нарушения целостности информации.

Разработанные математические модели обеспечения устойчивости функционирования РИС могут быть использованы на этапе проектирования, при эксплуатации и совершенствовании РИС (рис. 1).

На стадии проектирования РИС для построения процесса ХОД последовательно решаются задачи определения числа и месторасположения центров ХОД в РИС, оптимизации состава комплекса средств хранения данных, оптимизации распределения ПО функциональных задач по узлам сети, оптимизации распределения информационных массивов по центрам ХОД, оптимизации распределения резерва информационных массивов по центрам ХОД.

На этапе эксплуатации в случае, когда воздействие дестабилизирующих факторов имело негативные последствия для РИС, например, нарушение функционирования отдельных сегментов сети, разрушение информационных массивов, выход из строя средств хранения и обработки данных и т.п., для обеспечения сохранения работоспособности РИС могут последовательно решаться задачи оптимизации распределения ПО функциональных задач по узлам сети, резерва информационных массивов и информационных массивов по центрам ХОД. При этом перераспределению подлежат те элементы ПО, информационные массивы и их резерв, нормальная эксплуатация которых была нарушена при воздействии дестабилизирующего фактора. Это обеспечит устойчивость функционирования РИС путем реорганизации процесса ХОД.

Для обеспечения оперативности принятия решения по реорганизации процесса ХОД решение указанных задач целесообразно производить методом ветвей и границ с применением алгоритма предварительного определения порядка ветвления переменных. В качестве метода оценки границ решения использовать симплекс-метод, в качестве стратегии ветвления – глобально-поисковую или локально-избирательную стратегию. При решении указанных задач должны учитываться результаты размещения остальных ЭПО и информационных массивов и их резерва, а также результаты воздействия дестабилизирующих факторов на функционирование узлов сети, центров ХОД, каналов передачи данных.

На этапе совершенствования РИС с вводом в эксплуатацию новых комплексов средств хранения и обработки данных, развертыванием новых сегментов распределенной вычислительной системы математические модели обеспечения устойчивости функционирования РИС могут решаться как в полном объеме, так и выборочно.

Для решения формализованных задач возможно использование разработанного программного комплекса [15, 16], реализующего метод ветвей и границ по решению задач дискретной оптимизации с булевыми переменными с применением алгоритма предварительного определения порядка ветвления переменных на основе использования теории двойственности. Использование теории двойственности для определения порядка ветвления переменных в методе ветвей и границ позволяет сократить время решения разработанных математических моделей в среднем в 8 раз. Применение разработанного программного комплекса позволяет получить решение разработанных задач за время, приемлемое для их практического применения.

Таким образом, использование разработанного комплекса математических моделей позволит обеспечить устойчивое функционирование РИС на этапах проектирования, эксплуатации и совершенствования.

Литература

1. Повзнер Л.Д. Теория систем управления: учеб. пособие для вузов. М.: Изд-во МГТУ, 2002. 472 с.
2. Николаев В.И., Брук В.М. Системотехника: методы и приложения. Л.: Машиностроение, 1985. 199 с.
3. Есиков Д.О. Способы повышения устойчивости функционирования вычислительной системы // Молодежные инновации: сб. статей 7-й молодеж. науч.-практич. конф. Тула: Изд-во ТулГУ, 2012.
4. Есиков Д.О., Есиков О.В., Савватеев С.С., Струков А.В., Сухарев Е.М. Математические модели, методики и алгоритмы обеспечения устойчивости функционирования современных систем хранения и обработки данных // Нелинейный мир. 2013. № 9. С. 621–632.
5. Кристофидес Н. Теория графов. Алгоритмический подход. М.: Мир, 1978. 430 с.
6. Yesikov D.O. Increase of level of stability of functioning of systems of storage and data processing at the expense of realization of actions for ensuring safety of information // High Performance Computing 3rd Intern. Conference (HPC-UA 2013), pp. 416–420.
7. Михалевиц В.С., Волкович В.Л., Волошин А.Ф. Метод последовательного анализа в задачах линейного программирования большого размера // Кибернетика. 1981. № 4. С. 114–120.
8. Подиновский В.В., Ногин В.Д. Парето-оптимальные решения многокритериальных задач. М.: Физматлит, 2007. 256 с.
9. Есиков Д.О. Применение теории оптимальности по Парето для обоснования требуемых значений параметров надежности функционирования комплексов средств хранения информации // Труды Российского науч.-технич. об-ва радиотехники, электроники и связи им. А.С. Попова. 2014. Вып. XXXII. 325 с.
10. Кини Р.П., Райфа Х. Применение решений при многих критериях: предпочтения и замещения. М.: Радио и связь, 1981. 560 с.
11. Есиков Д.О. Математические модели построения подсистемы обеспечения сохранности информации в распределенных информационных системах // Труды Российского науч.-технич. об-ва радиотехники, электроники и связи им. А.С. Попова. 2014. Вып. XXXII. 325 с.
12. Воронин А.А., Морозов Б.И. Надежность информационных систем: учеб. пособие. СПб: Изд-во СПбГТУ, 2001.
13. Есиков Д.О., Абрамов П.И. Математическая модель обеспечения сохранности информации в распределенных информационных системах за счет резервирования информационных массивов // Труды Российского науч.-технич. об-ва радиотехники, электроники и связи им. А.С. Попова. 2014. Вып. XXXII. 325 с.
14. Алексеев О.Г. Комплексное применение методов дискретной оптимизации. М.: Наука, 1987. 248 с.
15. Есиков Д.О., Ивутин А.Н. Программа решения задач целочисленного линейного программирования с булевыми переменными. Свид. о гос. регистр. программы для ЭВМ № 2014619322.
16. Есиков Д.О. Программа распределенного решения задач целочисленного линейного программирования с булевыми переменными методом ветвей и границ. Свид. о гос. регистр. прог. для ЭВМ № 2014619324.

Abstract. The paper considers the ways to ensure distributed information systems sustainability. It formalizes the problems of ensuring sustainability of distributed information systems, which consist of: a mathematical optimization model for functional tasks software elements distribution according to network nodes; a mathematical optimization model for information resources distribution according to data storage and processing centers; a mathematical model for determination of a reasonable expenditure level to form data storage facilities; a mathematical optimization model of technical means for data storage and processing; a mathematical optimization model for information resources reserve

distribution on data storage and processing centers. The article shows that they belong to discrete optimization problems and gives their characteristics. It also proposes a procedure for applying a complex of mathematical models to ensure distributed information systems sustainability with indication of input and output data for each model. To solve the abovementioned problems the authors suggest using a software package, which implements the branch and bound method to solve discrete optimization problems with Boolean variables using the algorithm of branching variables predetermined order based on the duality theory. The duality theory in the branch and bound method significantly enhances the screening of unpromising options and reduces the time for solving problems compared with the traditional method on average in 8 times. The paper gives the special aspects of practical use of the developed mathematical model complex during design, maintenance and update of distributed information systems life cycle.

Keywords: distributed information system, mathematical models, discrete optimization, Pareto optimization, operational sustainability.

References

1. Povzner L.D. *Teoriya sistem upravleniya* [The Theory of Control Systems]. A manual for universities. Moscow, MEI Moscow State Mining Univ. Publ., 2002, 472 p.
2. Nikolaev V.I., Brooke V.M. *Sistemotekhnika: metody i prilozheniya* [System engineering: methods and applications]. Leningrad, Mashinostroenie Publ., 1985, 199 p.
3. Yesikov D.O. Ways to improve computer system sustainability. *Molodezhnye innovatsii: sb. statey 7-y molodezh. nauch.-praktich. konf.* [Proc. of the 7th Youth Scientific and Practical Conf. "Youth Innovation"]. Tula, TSU Publ., 2012.
4. Yesikov D.O., Yesikov O.V., Savvateev S.S., Strukov A.V., Sukharev E.M. Mathematical models, methods and algorithms to ensure the sustainability of modern storage systems and data processing. *Nelineyny mir* [Nonlinear World]. Moscow, 2013, no. 9, pp. 621–632 (in Russ.).
5. Christofides N. *Graph Theory. Algorithmic approach*. Academic Press, 1975, 415 p. (Russ. ed.: Moscow, Mir Publ., 1978, 430 p.).
6. Yesikov D.O. Increase of level of stability of functioning of systems of storage and data processing at the expense of realization of actions for ensuring safety of information. *3rd Int. Conf. "High Performance Computing" (HPC-UA 2013)*. Ukraine, Kiev, 2013, pp. 416–420.
7. Mikhalevich V.S., Volkovich V.L., Voloshin A.F. Sequential analysis in large linear programming tasks. *Kibernetika* [Cybernetics]. 1981, no. 4, pp. 114–120 (in Russ.).
8. Podinovskiy V.V., Nogin V.D. *Pareto-optimalnye resheniya mnogokriterialnykh zadach* [Pareto-optimal solutions for multi-criteria problems]. Moscow, Fizmatlit Publ., 2007, 256 p.
9. Yesikov D.O. Application of the Pareto optimality theory to justify the required parameters of data storage complexes reliability. *Trudy Ross. nauch.-tekhnich. ob-va radiotekhniki, elektroniki i svyazi im. A.S. Popova* [Proc. of the Russian Scientific and Technical Society of Radio Engineering, Electronics and Telecommunications n.a. A.S. Popov]. 2014, iss. 32, 325 p.
10. Keeney R.L., Raiffa H. *Decision-making with multiple criteria: preferences and value tradeoffs*. NY, Wiley Publ., 1976 (Russ. ed.: Moscow, Radio and svyaz Publ., 1981, 560 p.).
11. Yesikov D.O. Mathematical models of building an information integrity subsystem in distributed information systems. *Trudy Ross. nauch.-tekhnich. ob-va radiotekhniki, elektroniki i svyazi im. A.S. Popova* [Proc. of the Russian Scientific and Technical Society of Radio Engineering, Electronics and Telecommunications n.a. A.S. Popov]. 2014, iss. 32, 325 p.
12. Voronin A.A., Morozov B.I. *Nadezhnost informatsionnykh sistem* [Information system reliability]. Study guide, St.-Petersburg, SPbSTU Publ., 2001.
13. Yesikov D.O., Abramov P.I. A mathematical model for information integrity in distributed information systems through data arrays reservation. *Trudy Ross. nauch.-tekhnich. ob-va radiotekhniki, elektroniki i svyazi im. A.S. Popova* [Proc. of the Russian Scientific and Technical Society of Radio Engineering, Electronics and Telecommunications n.a. A.S. Popov]. 2014, iss. 32, 325 p.
14. Alekseev O.G. *Kompleksnoe primeneniye metodov diskretnoy optimizatsii* [Integrated application of discrete optimization]. Moscow, Nauka Publ., 1987, 248 p.
15. Yesikov D.O., Ivutin A.N. *Программа распределенного решения задач целочисленного линейного программирования с булевыми переменными методом ветвей и границ* [Program of solving integer linear programming problem with Boolean variables using the branch and bound method]. Software State Registration Certificate no. 2014619322.
16. Yesikov D.O. *Программа распределенного решения задач тселочисленного линейного программирования с булевыми переменными методом ветвей и границ* [Program distributed solving of integer linear programming problems with Boolean variables using the branch and bound method]. Software State Registration Certificate no. 2014619324.

УДК 519.68

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.142-147

РЕАЛИЗАЦИЯ ПРОЦЕССА ПОИСКА РЕШЕНИЯ ПО МОДИФИЦИРОВАННОМУ АЛГОРИТМУ РЕТЕ ДЛЯ НЕЧЕТКИХ ЭКСПЕРТНЫХ СИСТЕМ

*И.С. Михайлов, к.т.н., доцент, fr82@mail.ru;
Зо Мин Тайк, аспирант, zawgyi86@gmail.com
(Национальный исследовательский университет
«Московский энергетический институт»,
ул. Красноказарменная, 14, г. Москва, 111250, Е-250, Россия)*

В работе рассматриваются основные понятия теории нечетких продукционных экспертных систем. Нечеткие продукционные экспертные системы базируются на наборе правил, представленном в терминах лингвистических переменных. В качестве механизма нечеткого вывода предлагается разработанная модификация алгоритма Rete для нечеткой базы правил. Разработанная модификация обеспечивает ускорение процесса работы системы за счет однократного вычисления одинаковых условий в правилах, а также позволяет формулировать правила и заключения на ограниченном естественном языке. Разработанная формальная модель дерева решений модифицированного алгоритма Rete для нечеткой продукционной базы знаний состоит из множеств вершин-условий, вершин-следствий, отношений между вершинами и отношений для описания правил нечеткой экспертной системы. Созданный алгоритм обрабатывает правила нечеткой базы правил и преобразует их в формат формальной модели дерева решений модифицированного алгоритма Rete. На каждом этапе работы алгоритма выполняется построение нечетких оценок истинности вершин дерева решений с помощью нечетких операторов, что позволяет формулировать условия и следствия в базе правил, а также результаты работы алгоритма поиска решения на ограниченном естественном языке. Также одинаковые условия объединяются при построении дерева решений, что обеспечивает ускорение обработки дерева решений по сравнению с последовательным просмотром правил экспертной системы. Рассмотрен пример работы нечеткой продукционной экспертной системы, функционирующей на основе предложенной модификации алгоритма Rete, показана эффективность предложенного метода.

Ключевые слова: нечеткая экспертная система, нечеткая база правил, алгоритм Rete, модификация алгоритма Rete.

Использование экспертных систем для решения объемных, трудно формализуемых задач в различных предметных областях приобретает все большую актуальность. Эти задачи характеризуются, как правило, отсутствием или сложностью формальных алгоритмов решения, неполнотой и нечеткостью исходной информации, нечеткостью достигаемых целей [1]. Данные особенности обусловили необходимость использования знаний в процессе решения этих задач и разработки экспертных систем, осуществляющих сбор и управление этими знаниями, принимающими решения об оптимальном способе достижения целей в условиях неполноты и нечеткости. При этом знания должны поступать в систему от человека-эксперта в предметной области. Знания человека-эксперта о решении задач в таких условиях также имеют нечеткий характер. Для их формализации успешно применяется аппарат теории нечетких множеств и нечеткой логики [2]. Нечеткие понятия в данном случае формализуются в виде нечетких и лингвистических переменных, а нечеткость действий в процессе принятия решения – в виде нечетких алгоритмов. Экспертные системы, способные формально представлять нечеткую информацию и обрабатывать ее в рамках нечетких алгоритмов, получили название нечетких экспертных систем [3]. Наиболее широко применяемой моделью знаний экспертных систем является продукционная модель в силу своей про-

стоты обработки и понятности конечному пользователю [4]. Нечеткие экспертные системы базируются на наборе правил, в которых используются лингвистические переменные и нечеткие отношения для описания состояния и поведения исследуемого объекта [5]. Правила, представленные в таком виде, наиболее приближены к естественному языку, поэтому нет необходимости в услугах отдельного специалиста – инженера по знаниям для создания и редактирования правил [6]. Они могут быть отредактированы самим экспертом практически без специальной подготовки [7].

В данной работе поставлена задача разработки и реализации модифицированного алгоритма Rete для нечеткой продукционной базы правил, а также разработки нечеткой продукционной экспертной системы, механизм вывода которой будет функционировать на основе созданной модификации этого алгоритма. Необходимо создать формальную модель дерева решений модифицированного алгоритма Rete. При работе решателя нечеткой продукционной экспертной системы все правила из базы правил применяются последовательно. При этом заново анализируются все условия, которые содержатся в данных правилах. Разрабатываемый алгоритм позволит проверять каждое условие только один раз. Таким образом будет обеспечено ускорение работы решателя нечеткой ЭС и, следовательно, быстрее будет найдено решение.

Нечеткая экспертная система с использованием формальной модели дерева решений модифицированного алгоритма Rete

Нечеткая экспертная система использует для вывода решения вместо булевой логики совокупность нечетких функций принадлежности и правил. Поведение исследуемой системы описывается на ограниченном естественном языке в терминах лингвистических переменных [8]. Входные и выходные параметры системы рассматриваются как лингвистические переменные, а описание процесса задается набором правил [9]. Формальная модель базы правил разработанной экспертной системы имеет вид

$$\begin{aligned} L_1 : A_{11} \text{ и/или } A_{21} \text{ и/или } \dots \text{ и/или } A_{1m} &\rightarrow B_{11} \\ \text{и/или } \dots \text{ и/или } B_{1n}, \\ L_2 : A_{21} \text{ и/или } A_{22} \text{ и/или } \dots \text{ и/или } A_{2m} &\rightarrow B_{21} \\ \text{и/или } \dots \text{ и/или } B_{2n}, \\ &\dots \\ L_k : A_{k1} \text{ и/или } A_{k2} \text{ и/или } \dots \text{ и/или } A_{km} &\rightarrow B_{k1} \\ \text{и/или } \dots \text{ и/или } B_{kn}, \end{aligned} \quad (1)$$

где A_{ij} , $i = 1, 2, \dots, k$, $j = 1, 2, \dots, m$ – нечеткие высказывания, определенные на значениях входных лингвистических переменных; B_{ij} , $i = 1, 2, \dots, k$, $j = 1, 2, \dots, n$ – нечеткие высказывания, определенные на значениях выходных лингвистических переменных. Эта совокупность правил носит название нечеткой базы знаний [3].

В общем случае нечеткий вывод решения выполняется за четыре этапа: фаззификация, непосредственный нечеткий вывод, композиция и дефаззификация. Этап фаззификации заключается в преобразовании с помощью функций принадлежности μ точных входных данных в нечеткие значения лингвистических переменных. На этапе непосредственного нечеткого вывода вычисляется значение истинности для условий каждого правила по правилам вычисления Т-норм, Т-конорм и отрицаний. Вычисления осуществляются на основе набора правил нечеткой базы знаний. Этап композиции состоит в формировании значений выходных лингвистических переменных для каждого сработавшего правила. На этапе дефаззификации выполняется преобразование нечетких значений выходных лингвистических переменных в точные значения.

В данной работе вычисления Т-норм и Т-конорм осуществляются по правилам $T_M(x, y) = \min\{x, y\}$ и $\perp_M(x, y) = \max\{x, y\}$. Отрицание утверждения вычисляется по правилу классического отрицания $\neg(x) = 1 - x$.

В настоящее время алгоритм Rete является эффективным алгоритмом сопоставления с образцом для продукционных экспертных систем [10]. Rete стал основой многих популярных экспертных систем, включая CLIPS, Jess, Drools, BizTalk Rules Engine и Soar. Классический алгоритм работы экс-

пертных систем заключается в проверке применимости каждого правила вывода к каждому факту базы знаний при необходимости его выполнения и переходе к следующему правилу с возвратом в начало базы знаний в случае исчерпания всех правил [11]. Даже для небольшого набора правил и фактов такой метод работает неприемлемо медленно. Алгоритм Rete обеспечивает более высокую эффективность [12]. При использовании Rete экспертная система выполняет формирование специального графа, узлам которого соответствуют части условий правил. Путь от корня до листа образует полное условие некоторой продукции.

В процессе работы каждый узел хранит список фактов, соответствующих условию. При добавлении или модификации факта он обрабатывается графом, при этом отмечаются узлы, условиям которых данный факт соответствует. При выполнении полного условия правила, когда система достигает листа графа, правило выполняется. В большинстве случаев скорость работы систем с использованием алгоритма Rete возрастает на порядки. Однако данный алгоритм реализован для экспертных систем с классическими продукционными правилами. В работе рассматривается модификация алгоритма Rete для работы с нечеткой базой правил.

Граф модификации Rete-алгоритма для нечетких экспертных систем формируется таким образом, что в каждом случае проверяется не точное значение условия правила, а значения лингвистических переменных в данном правиле. Формальная модель дерева решений будет иметь вид

$$M = (X, R, P, Y), \quad (2)$$

где X – множество вершин-условий графа; Y – множество вершин-следствий графа; R – множество отношений между вершинами; P – множество отношений для описания правил нечеткой экспертной системы. Каждая вершина-условие представляет собой условия из правил нечеткой базы знаний: $X = \{ \langle l_i, k_i, Tx_i, z_i \rangle \}$, где l_i – название лингвистической переменной; k_i – отношение сравнения, $k_i \in K$, где $K = \{=, \neq\}$; Tx_i – значение лингвистической переменной, то есть нечеткая переменная из ее терм-множества; z_i – значение степени уверенности μ в том, что рассматриваемое условие l_i, k_i, Tx_i истинно, $i = 1, \dots, n$, где n – количество условий в базе знаний. Каждая вершина-следствие представляет собой присваивание определенной лингвистической переменной ее значения: $Y = \{ \langle l_i = Ty_i, z_i \rangle \}$, где l_i – название лингвистической переменной; Tx_i – значение лингвистической переменной, то есть нечеткая переменная из ее терм-множества; z_i – значение степени уверенности μ в том, что рассматриваемое следствие $l_i = Tx_i$ истинно, $i = 1, \dots, n$, где n – количество условий в базе знаний.

Множество R отношений между вершинами из объединения множеств X и Y имеет вид: $R \subset ((X \cup Y) \times (X \cup Y))$ или $R: (X \cup Y) \rightarrow (X \cup Y)$. R может прини-

мать следующие значения: $R = \{T, \perp, \neg\}$. Для операций Т-норм, Т-конорм и отрицания приняты следующие обозначения: x_i и x_j – условия из $(X \cup Y)$, $x_i \in (X \cup Y)$, $x_j \in (X \cup Y)$; z_i – значение степени уверенности в истинности условия $x_i = \langle l_i, k_i, Tx_i, z_i \rangle$; z_j – значение степени уверенности в истинности условия $x_j = \langle l_j, k_j, Tx_j, z_j \rangle$; z_k – значение степени уверенности в истинности результатов вычисления нормы, конормы или отрицания. Отношение Т-нормы определяется следующим образом: $z_k = T(x_i, x_j)$. Значения вычисляются по правилу $T(x_i, x_j) = \min\{z_i, z_j\}$. Отношение Т-конормы определяется следующим образом: $z_k = \perp(x_i, x_j)$. Значения вычисляются по правилу $\perp(x_i, x_j) = \max\{z_i, z_j\}$. Отношение отрицания определяется следующим образом: $z_k = \neg(x_i)$. Значение вычисляется по правилу $\neg(x_i) = 1 - z_i$.

Множество отношений P для описания правил нечеткой экспертной системы имеет вид $P \subset ((X \cup Y) \times Y)$ или $P: (X \cup Y) \rightarrow Y$. В рассматриваемой модели P может принимать одно значение: $P = \{ \rightarrow \}$, где $\rightarrow$ связывает значения в левой части и вершины-следствия в правой части и означает, что при истинности значений в левой части (то есть при степени уверенности в них $> 0,5$) выполняются утверждения, которые расположены в правой части.

Рассмотрим процесс преобразования нечеткой базы правил в дерево решений, представленное в формате разработанной формальной модели.

Модификация алгоритма Rete для формирования дерева решений нечеткой продукционной базы правил

Рассмотрим модифицированный алгоритм Rete, который был адаптирован для работы с нечеткими продукционными правилами. Данный алгоритм обрабатывает правила нечеткой базы правил и преобразует их в формат формальной модели дерева решений модифицированного алгоритма Rete. Рассмотрим схему данного алгоритма.

Входные данные алгоритма: правила из базы правил.

1. Начало алгоритма.
2. Установить $i=1$.
3. Извлечь из базы правил следующее непомеченное условие и сформировать вершину x_i .
4. Просмотреть базу правил и пометить все позиции в правилах, на которых встречается данное условие.
5. Сформировать вершины для описания отношений из $R \{T, \perp, \neg\}$, с которыми связано x_i . Установить отношения между данными вершинами.
6. Если в базе правил не закончились непомеченные условия, увеличить i на 1 и осуществить переход к шагу 3.
7. Установить $j=1$.

8. Извлечь из базы правил следующее непомеченное следствие и сформировать вершину y_j .

9. Просмотреть базу правил и пометить все позиции в правилах, на которых встречается данное следствие.

10. Сформировать вершину для описания отношения из $P \{ \rightarrow \}$, с которым связано y_j . Установить отношения между данными вершинами.

11. Если вершины И, ИЛИ, НЕ, с которыми связаны отношения из P , не сформированы, сформировать их.

12. Если вершины И, ИЛИ, НЕ, с которыми связаны отношения из P , уже сформированы, установить связи с ними.

13. Если в базе правил не закончились непомеченные следствия, увеличить j на 1 и осуществить переход к шагу 9.

14. Конец алгоритма.

Выходные данные: дерево решений, построенное на основе правил.

Экспертная система осуществляет фаззификацию входных данных и формирует значения лингвистических переменных. Затем выполняется подстановка полученных значений в созданное дерево решений, которое формирует результат работы экспертной системы. Далее может быть выполнена дефаззификация результата для получения числовых характеристик.

Пример использования разработанного алгоритма

Рассмотрим пример использования предложенного алгоритма для нечеткой экспертной системы, содержащей набор правил, состоящих из лингвистических переменных. В системе рассматриваются следующие входные лингвистические переменные:

TP = «Температура пациента» = {«низкая», «нормальная», «высокая»};

NL = «Количество лейкоцитов в крови» = {«низкое», «нормальное», «высокое»};

LS = «Уровень сахара в крови» = {«низкий», «нормальный», «высокий»}.

Выходные лингвистические переменные:

DN = «Дозировка лекарства N » = {«низкая», «стандартная», «высокая»};

DM = «Дозировка лекарства M » = {«низкая», «стандартная», «высокая»}.

Также система содержит следующий набор правил:

$((TP = \text{«высокая»}) \text{ и } (NL = \text{«высокое»})) \text{ и } (LS = \text{«низкий»}) \rightarrow (DN = \text{«высокая»});$

$((TP = \text{«высокая»}) \text{ и } (NL = \text{«высокое»})) \text{ и } (LS = \text{«нормальный»}) \rightarrow (DN = \text{«низкая»});$

$((TP = \text{«низкая»})) \rightarrow (DM = \text{«низкая»});$

$((TP = \text{«высокая»}) \text{ и } (NL = \text{«высокое»})) \rightarrow (DM = \text{«высокая»}).$

Рассмотрим работу экспертной системы на примере пациента со следующими симптомами: температура = 37,5 °С, количество лейкоцитов в крови = $9,5 \cdot 10^9/\text{л}$, уровень сахара в крови = 4,8 ммоль/л.

В начале работы алгоритма следует выполнить фаззификацию исходных данных на основе значений заданных функций принадлежности:

– температура = 37,5 °С, тогда

$M_{TP_высокая}(37,5\text{ °С})=0,9$,

$M_{TP_низкая}(37,5\text{ °С})=0,01$,

$M_{TP_нормальная}(37,5\text{ °С})=0,2$;

– количество лейкоцитов в крови = $9,5 \cdot 10^9/\text{л}$, тогда при норме $5,5\text{--}8,8 \cdot 10^9/\text{л}$ получим:

$M_{NL_высокое}(9,5 \cdot 10^9/\text{л})=0,7$,

$M_{NL_низкое}(9,5 \cdot 10^9/\text{л})=0,02$,

$M_{NL_нормальное}(9,5 \cdot 10^9/\text{л})=0,4$;

– уровень сахара в крови = 4,8/л, тогда при норме 3,3–5,5 ммоль/л получим:

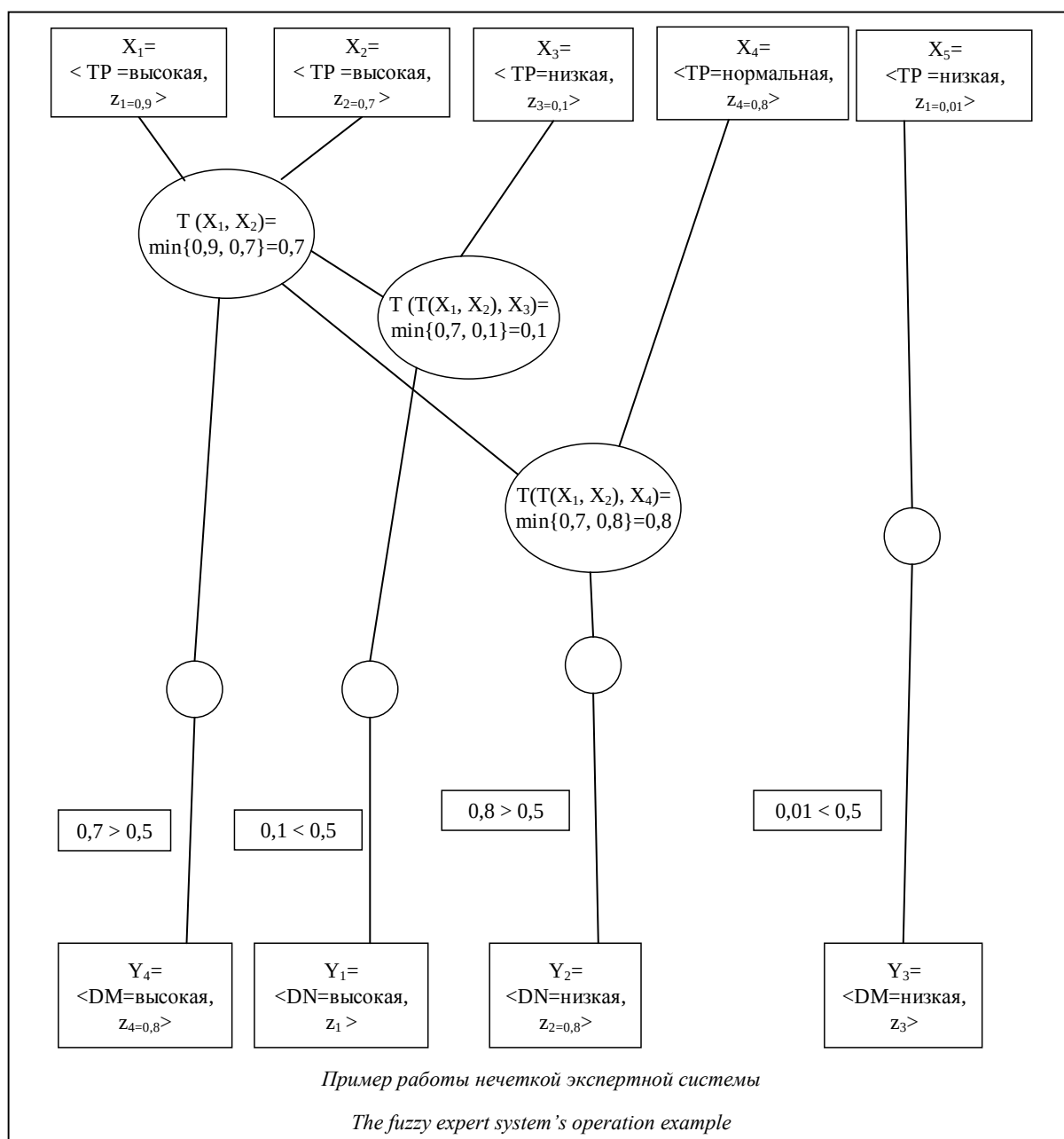
$M_{LS_высокий}(4,8\text{ ммоль/л})=0,3$,

$M_{LS_низкий}(4,8\text{ ммоль/л})=0,1$,

$M_{LS_нормальный}(4,8\text{ ммоль/л})=0,8$.

Преобразуем представленную базу правил в дерево решений модифицированного алгоритма Rete и выполним этап вывода для заданных исходных данных. Схема формирования дерева решений для рассматриваемого примера представлена на рисунке.

В начале алгоритма устанавливается $i=1$. Из базы правил извлекается следующее непомеченное условие « TP = высокая» и формируется вершина X_i . Осуществляется просмотр базы правил и отмечаются все позиции в правилах, на которых встреча-



ется данное условие. Формируется вершина для описания отношений из R , с которыми связана вершина X_i : $T(X_1, X_2)$. Если в базе правил не закончились непомянутые условия, то i увеличивается на 1 и процесс формирования вершин для описания отношений из R продолжается. Если новое условие является второй частью одного из отношений из R , то оно связывается с уже сформированной вершиной для первой части данного отношения. Так будут сформированы $T(T(X_1, X_2), X_3)$, $T(T(X_1, X_2), X_4)$.

Затем устанавливается $j=1$. Из базы правил извлекается следующее непомянутое следствие « DN = высокая» и формируется вершина Y_j . Осуществляется просмотр базы правил и отмечаются все позиции в правилах, на которых встречается данное следствие. Формируется вершина для описания отношений из P , с которыми связано Y_1 . Устанавливается отношение между данными вершинами. Если вершины, с которыми связаны отношения из P , уже сформированы, устанавливаются связи с ними. Связываем вершины $T(X_1, X_2)$ и Y_1 . Если их нет, то данные вершины формируются. Если в базе правил не закончились непомянутые условия, то j увеличивается на 1 и процесс построения следствий продолжается.

Таким образом, после работы механизма вывода нечеткой экспертной системы будет сформирован следующий ответ: «Для пациента со следующими симптомами: температура = 37,5 °C, количество лейкоцитов в крови = $9,5 \cdot 10^9$ /л, уровень сахара в крови = 4,8/л, назначить высокую дозировку лекарства M с уверенностью 0,7, низкую дозировку лекарства N с уверенностью 0,8».

Затем выполняется этап дефазификации. На данном этапе значения полученных лингвистических переменных преобразуются в числовые характеристики.

Например, лингвистическая переменная DN имела следующие значения: DN = «Дозировка лекарства N » = {«низкая», «стандартная», «высокая»}; и нечеткая переменная «низкая» соответствовала 1 таблетке, «стандартная» – 2 таблеткам, «высокая» – 3 таблеткам. Тогда построенное решение соответствует дозировке 1 таблетка с уверенностью 0,9.

В рассматриваемом примере видно, что несколько правил содержат одинаковые условия. При классической схеме проверки правил данные условия проверялись бы в каждом правиле отдельно. Предложенный алгоритм позволяет вычислять нечеткие условия только один раз, что дает существенный выигрыш в скорости принятия решения нечеткой экспертной системой. В данном примере при классической схеме обработки правил выполнялась бы проверка десяти условий, а при использовании модифицированного алгоритма Rete осуществилась проверка только пяти условий. Таким образом наглядно показана эффективность предложенного метода. Следует отметить, что при ис-

пользовании предложенного метода в реальных экспертных системах сокращение объема вычислений зависит от количества повторяющихся условий в правилах базы знаний.

При работе классического алгоритма Rete выделяются общие условия в различных правилах продукционной экспертной системы. Затем на их основе выполняется построение дерева решений. То есть каждая вершина алгоритма может быть либо истинна, либо ложна. Модификация алгоритма Rete отличается от классического алгоритма тем, что он применяется для нечетких переменных. В связи с этим на каждом этапе работы алгоритма выполняется построение нечетких оценок истинности вершин дерева решений с помощью операторов Т-норм, Т-конорм и отрицания. Это позволяет формулировать условия и следствия в базе правил на ограниченном естественном языке. Заключение формируются алгоритмом также на ограниченном естественном языке.

Необходимо отметить, что одинаковые условия объединяются при построении дерева решений на основе правил базы знаний. Это дает ускорение обработки дерева решений по сравнению с последовательным просмотром правил экспертной системы.

В заключение отметим, что в работе получены следующие результаты. Выполнено исследование основных понятий нечеткой логики и процесса создания и функционирования нечетких экспертных систем продукционного типа, а также изучены процессы и технологии разработки БД. Полученные навыки применены на практике при создании БД для тестовых примеров. Кроме того, разработана формальная модель дерева решений модифицированного алгоритма Rete для нечеткой продукционной базы знаний, предложена модификация алгоритма Rete для формирования дерева решений нечеткой продукционной базы правил и рассмотрен пример использования предложенного алгоритма. Показана его эффективность. Авторами разработана экспертная система, которая позволяет формировать решение для подключаемой базы правил. Экспертная система преобразует нечеткую продукционную базу знаний в дерево решений Rete и в дальнейшем формирует заключение на основе обработки данной структуры.

Литература

1. Прикладные нечеткие системы; [под ред. Тэрано Т., Асаи К., Сугэно М.]. М.: Мир, 1993. 368 с.
2. Искусственный интеллект. В 3 кн. Кн. 2: Модели и методы: справочник. М.: Радио и связь, 1990. 304 с.
3. Заде Л. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.: Мир, 1976. 166 с.
4. Вагин В.Н., Еремеев А.П. Некоторые базовые принципы построения интеллектуальных систем поддержки принятия решений реального времени // Изв. РАН. ТиСУ, 2001. № 6. С. 114–123.
5. Заде Л. От обработки чисел к обработке слов – от манипулирования измерениями к манипулированию восприятием //

Междунар. журн. прикладной матем. и компютер. науки. 2002. Т. 12. № 3. С. 307–324.

6. Аверкин А.Н., Хорошевский В.Ф. Нечеткое отношение моделирования и его использование для классификации и аппроксимации в нечетких лингвистических пространствах // Изв. АН СССР: Техническая кибернетика. 1982. № 2. С. 21.

7. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. СПб: Питер, 2001. 384 с.

8. Аверкин А.Н., Батыршин И.З., Блишун А.Ф., Силов В.Б., Тарасов В.Б. Нечеткие множества в моделях управления и искусственного интеллекта; [под ред. Д.А. Поспелова]. М.: Наука. Физматлит, 1986. 312 с.

9. Блюмин С.Л., Шуйкова И.А., Сараев П.В., Черпаков И.В. Нечеткая логика: алгебраические основы и приложения: монография. Липецк: Изд-во ЛЭГИ, 2002. 111 с.

10. Форги Ч. Rete: быстрый алгоритм для многошаблонных/многообъектных задач сопоставления с образцом // Искусственный интеллект. 1982. № 19. С. 17–37.

11. Рассел С., Норвиг П. Искусственный интеллект: современный подход. М.: Вильямс, 2006. 1407 с.

12. Вакин В.В., Корухова Л.С., Любимский Э.З., Малышко В.В. Ассоциативные методы планирования решений сложных задач. М.: Препр. ИПМ им. М.В. Келдыша РАН, 1997. № 81.

DOI: 10.15827/0236-235X.112.142-147

Received 09.09.15

FINDING SOLUTIONS BY THE MODIFIED RETE ALGORITHM FOR FUZZY EXPERT SYSTEMS

Mikhailov I.S., Ph.D. (Engineering), Associate Professor;

Zaw Min Htike, Postgraduate Student, zawgyi86@gmail.com

(National Research University "MPEI", Krasnokazarmennaya St. 14, Moscow, 111250, Russian Federation)

Abstract. The paper considers the basic concepts of fuzzy production expert systems. Fuzzy production expert systems are based on a set of rules presented in terms of linguistic variables. The authors suggest the developed Rete algorithm modification for a fuzzy rule base as a fuzzy inference tool. This modification accelerates systems operation due to a single computing of the same conditions in the rules. It also formulates the rules and conclusions in the limited natural language. The modified Rete algorithm formal decision tree model for a fuzzy production knowledge base consists of a set of vertex-conditions, vertex-solutions, the relationship between vertices and relations to describe the fuzzy expert system rules. The created algorithm processes the rules from the fuzzy rule base and converts them into the decision tree modified Rete algorithm formal model. Rete algorithm modification is different from a classical algorithm as it is used for fuzzy variables. Therefore, each stage of the algorithm includes building the decision tree vertices fuzzy truth values using fuzzy operators. This allows formulating the conditions and consequences in the rule base, as well as the solutions in the limited natural language. The same conditions are combined during decision tree construction. It accelerates decision tree processing comparing to sequential viewing of expert system rules. The paper describes an operating example of the production fuzzy expert system, which works on the basis of the proposed Rete algorithm modification. It also displays the effectiveness of the proposed method.

Keywords: fuzzy expert system, fuzzy rule base, Rete algorithm, Rete algorithm modification.

References

1. Terano T., Asai K., Sugeno M. (Eds.). *Applied fuzzy systems*. 1991 (Russ. ed.: Moscow, Mir Publ., 1993, 368 p.).
2. *Iskusstvennyy intellekt. Kn. 2: Modeli i metody* [Artificial Intelligence. Vol. 2: Models and methods]. Handbook. Moscow, Radio i svyaz Publ., 1990, 304 p.
3. Zadeh L. *Ponyatie lingvisticheskoy peremennoy i ego primeneniye k prinyatiyu priblizhennykh resheniy* [The Linguistic Variable Concept and its Application to Approximate Solutions]. Moscow, Mir Publ., 1976, 166 p.
4. Vagin V.N., Eremeev A.P. Some Basic Principles of Design of Intelligent Systems for Supporting Real-Time Decision Making. *Izvestia RAN. Teoriya i systemy upravleniya* [Journ. of Computer and Systems Sciences Int.]. 2001, vol. 40, no. 6, pp. 953–961.
5. Zadeh L. From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions. *Int. Journ. of Applied Math and Computer Science*. 2002, vol. 12, no. 3, pp. 307–324.
6. Averkina A.N. Fuzzy relation of modeling and its use to classify and approximate in fuzzy linguistic spaces. *Izv. AN SSR, Ser. Tekhn. kibernetika* [News of AS SSR, Series Technical Cybernetics]. 1982, no. 2, p. 21.
7. Gavrilova T.A., Khoroshevsky V.F. *Bazy znaniy intellektualnykh sistem* [Knowledge Bases in Intelligent Systems]. Peter Publ., 2001, 384 p.
8. Averkina A.N., Batoryshin I.Z., Blishun A.F., Silov V.B., Tarasov V.B. *Nechetkie mnozhestva v modelyakh upravleniya i iskusstvennogo intellekta* [Fuzzy Sets in Management Models and Artificial Intelligence]. D.A. Pospelov (Ed.) Moscow, Nauka, Fizmatlit Publ., 1986, 312 p.
9. Blyumin S.L., Shuykova I.A., Saraev P.V., Cherpakov I.V. *Nechetkaya logika: algebraicheskie osnovy i prilozheniya* [Fuzzy Logic: Algebraic Fundamentals and Applications]. Monograph. Lipetsk, LEGI Publ., 2002, 111 p.
10. Forgy Ch. Rete: the Fast Algorithm for Multi-pattern/Multi-object Pattern Match Problems. *Iskusstvennyy intellekt* [Artificial Intelligence]. 1982, no. 19, pp. 17–37.
11. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995, 960 p. (Russ. ed.: Villiams Publ., 2006, 1407 p.).
12. Vakin V.V., Korukhova L.S., Lyubimsky E.Z., Malysheko V.V. *Assotsiativnye metody planirovaniya resheniy slozhnykh zadach* [Associative planning solutions methods for complex problems]. Moscow, Applied Mathematics Institute Preprint, Keldysh RAS, 1997, no. 81.

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА МУРАВЬИНЫХ КОЛОНИЙ ДЛЯ КРИПТОАНАЛИЗА БЛОЧНЫХ КРИПТОСИСТЕМ

(Работа выполнена при финансовой поддержке РФФИ, проекты №№ 14-01-00634, 15-05-00129)

Ю.О. Чернышев, д.т.н., профессор, myvnn@list.ru;

А.С. Сергеев, к.т.н., докторант, sergeev00765@mail.ru;

А.Н. Рязанов, аспирант, alexandr_r89@mail.ru

*(Донской государственный технический университет,
пл. Гагарина, 1, г. Ростов-на-Дону, 344000, Россия);*

С.А. Капустин, к.т.н., доцент, skappa@yandex.ru

(Филиал Военной академии связи, ул. Красина, 4, г. Краснодар, 350035, Россия)

В статье рассматривается возможность параллельной реализации алгоритмов муравьиных колоний для криптоанализа блочных криптосистем. Отмечена актуальность нового научного направления «природные вычисления», приведена структурная схема криптоанализа стандарта DES с использованием метода муравьиных колоний. Приводится описание параллельной версии алгоритма криптоанализа на основе информационно-логической граф-схемы, матриц следования, логической несовместимости и независимости. На основе методики определения числа процессоров, сущность которой заключается в нахождении максимального множества взаимно независимых операторов в матрице несовместимости, последовательном проведении фиктивных связей в информационно-логическом графе, не увеличивающих длину критического пути, определено минимальное число процессоров, необходимых для реализации алгоритма криптоанализа. Отмечается, что отличительной особенностью применения биоинспирированных методов криптоанализа является возможность использования самого алгоритма шифрования (или расшифрования) в качестве целевой функции для оценки пригодности ключа, определенного с помощью генетических операций. Вследствие этого при использовании биоинспирированных методов криптоанализа процесс определения секретного ключа (например, при криптоанализе 2-го типа) зависит не столько от сложности шифрующих преобразований, сколько от самого биоинспирированного метода, который должен обеспечивать достаточное разнообразие генерации ключей, что свидетельствует об актуальности задачи исследования возможности применения биоинспирированных алгоритмов (в частности, методов генетического поиска) для криптоанализа блочных криптосистем.

Ключевые слова: криптоанализ, муравьиный алгоритм, матрица логической несовместимости, матрица независимости, множества взаимно независимых операторов.

В последние годы интенсивно разрабатывается новое научное направление «природные вычисления», объединяющее математические методы, в которых заложен принцип природных механизмов принятия решений. К таким методам относятся методы моделирования отжига, эволюционного моделирования, генетические алгоритмы, методы эволюционной адаптации, алгоритмы роевого интеллекта, муравьиные алгоритмы. В моделях и алгоритмах эволюционных вычислений ключевым элементом является построение начальной модели и правил, по которым она может изменяться (эволюционировать). Были предложены разнообразные схемы эволюционных вычислений, в том числе генетический алгоритм, генетическое программирование, эволюционные стратегии, эволюционное программирование. Общие концепции и методологический подход к построению эволюционных вычислений, основанных на природных системах, а также основные гипотезы, закономерности и положения концепции эволюционных вычислений отмечены в [1, 2]. В настоящее время известно применение генетических алгоритмов для оптимизации широкого круга задач, в том числе задач криптоанализа. В [3–10] рассматривались методы организации криптографических атак на традиционные симметричные криптосистемы, использующие

шифры перестановки и замены, а также на блочные криптосистемы с использованием методов эволюционной оптимизации и генетического поиска.

Недостатком генетических алгоритмов является наличие «слепого» поиска. Это в общем случае приводит к генерации решений с нарушениями, что увеличивает время поиска и требует дополнительного контроля, к генерации большого количества одинаковых решений, генерации большого количества плохо приспособленных решений, что в общем случае может привести к попаданию в локальный оптимум [11]. Поэтому представляет интерес применение эвристических методов, инспирированных природными системами, в которых осуществляется поэтапное построение решения задачи (то есть добавление нового оптимального частичного решения к уже построенному частичному оптимальному решению). К методам данного вида относятся и муравьиные алгоритмы, идея которых состоит в моделировании поведения муравьев, связанного с их способностью быстро находить кратчайший путь от муравейника к источнику пищи. Как отмечено в [12], колония муравьев рассматривается как многоагентная система, в которой каждый муравей-агент функционирует автономно на основе простых правил. Однако в противоположность примитивному поведению отдельных аген-

тов поведение всей колонии оказывается достаточно разумным. Исследования в этой области проводятся с середины 90-х годов, и на сегодняшний день известно их применение к задаче о коммивояжере, квадратичной задаче о назначениях, к задачам о раскраске графа, определения оптимальных маршрутов в коммутационных сетях, маршрутизации транспортных средств, борьбы с вредоносным ПО. В [13] рассмотрен возможный подход к реализации криптоанализа блочных алгоритмов шифрования и показано сведение криптоанализа к классической задаче о назначении, решаемой с помощью алгоритма муравьиных колоний.

В настоящее время актуальной является задача криптоанализа блочных криптосистем, так как переход к блочному шифрованию открывает дополнительные возможности для повышения стойкости криптоалгоритмов. Сведения по криптостойкости ряда блочных алгоритмов, в частности, к линейному и дифференциальному криптоанализу, а также результаты его применения к алгоритму DES приведены в [14, 15]. Описание некоторых блочных стандартов шифрования (в том числе стандарта DES) с введенными терминами и обозначениями дано в [15–17].

Отличительной особенностью применения биоинспирированных методов криптоанализа является возможность использования самого алгоритма шифрования (или расшифрования) в качестве целевой функции для оценки пригодности ключа, определенного с помощью генетических операций. Это особенно существенно при реализации криптоанализа блочных алгоритмов шифрования, в которых применяется многократная обработка блоков текста и на каждом цикле данные преобразуются при участии вспомогательного ключа, сформированного из секретного ключа. Как отмечено в [15], выбор числа циклов определяется требованиями криптостойкости, то есть чем больше циклов, тем выше криптостойкость и больше времени требуется на шифрование (расшифрование). В связи с этим процесс криптоанализа, связанный с какими-либо преобразованиями самого алгоритма шифрования, может оказаться затруднительным и актуальной становится задача исследования возможности применения биоинспирированных алгоритмов для криптоанализа блочных криптосистем, поскольку при использовании данных алгоритмов процесс определения секретного ключа (например при криптоанализе 2-го типа) зависит не столько от сложности шифрующих преобразований, сколько от самого биоинспирированного метода, который должен обеспечивать достаточное разнообразие генерации ключей.

Увеличение производительности «природных» алгоритмов наряду с параллельной реализацией на «низшем» уровне (то есть при параллельной реализации самого алгоритма криптоанализа, используемого для оценки целевой функции пригодности

ключа) возможно также при распараллеливании на «высшем» уровне (то есть при реализации самого «природного» алгоритма, используемого для формирования популяции решений, их оценки и проведения множества операций, имитирующих эволюцию живой природы).

Чрезвычайно важным свойством эволюционных стратегий является их естественный внутренний параллелизм. При этом по сравнению с градиентными методами различие во времени расчета целевой функции при различных значениях параметров не оказывает влияния на эффективность распараллеливания [18]. Основные технологии распараллеливания алгоритмов (глобальные параллельные алгоритмы, островная модель, клеточный алгоритм) описаны в [18, 19], методы разработки параллельных программ – в [20, 21].

Разработку параллельной версии алгоритма рассмотрим на примере стандарта DES. Структурная схема криптоанализа с использованием метода муравьиных колоний и структурная схема цикла алгоритма DES приведены на рисунках 1, 2 и в работе [13].

На основе структурной схемы алгоритма муравьиных колоний и информационно-логической граф-схемы алгоритма DES, приведенных в [5, 6], получим информационно-логическую граф-схему алгоритма криптоанализа (рис. 3). Связи по управлению показаны на рисунке двойной линией, по информации – одинарной. Для упрощения будем предполагать, что число ключей K_i равно 2 (блоки 1, 2), вычисление целевой функции R для ключей K_1, K_2 производится на одной группе процессоров, реализующих блоки 3–23, число комбинаций $i_k k_k$ и число $m \cdot d$ маршрутов, образующих расширенную популяцию, равны 2, оценка их целевой функции пригодности проводится в одном блоке. Применительно к структурной схеме алгоритма DES будем предполагать, что в выходном блоке формируется блок исходного текста, на входной блок поступает блок шифртекста (поскольку процессы шифрования и расшифрования выполняются аналогично за исключением порядка использования ключа и матриц перестановок IP и IP^{-1}).

Для данного графа введем в рассмотрение матрицу следования S . Элемент $S_{ij} = *$, если существует связь по управлению ($j \Rightarrow i$), и $S_{ij} = 1$, если существует связь по информации ($j \rightarrow i$). Данная матрица будет содержать следующие ненулевые элементы:

$S(3, 1) = S(3, 2) = S(5, 3) = S(6, 5) = S(7, 5) = S(8, 1) = S(8, 2) = S(9, 8) = S(10, 8) = S(11, 8) = S(12, 8) = S(13, 8) = S(14, 8) = S(15, 8) = S(16, 8) = 1$; $S(9, 4) = S(10, 4) = S(11, 4) = S(12, 4) = S(13, 4) = S(14, 4) = S(15, 4) = S(16, 4) = *$;

$S(17, 7) = S(17, 9) = S(17, 10) = S(17, 11) = S(17, 12) = S(17, 13) = S(17, 14) = S(17, 15) = S(17, 16) = S(18, 17) = S(19, 18) = S(20, 5) = S(20, 19) = S(21, 6) = S(21, 20) = S(24, 22) = S(28, 26) = S(27, 29) = S(28, 30) = S(29, 31) = S(32, 1) = S(32, 2) = S(32, 30) =$

$S(32, 31) = S(33, 1) = S(33, 2) = S(33, 30) = S(33, 31) = S(34, 32) = S(34, 33) = S(35, 34) = 1;$
 $S(22, 21) = S(23, 21) = S(25, 24) = S(26, 24) = S(27, 24) = *.$

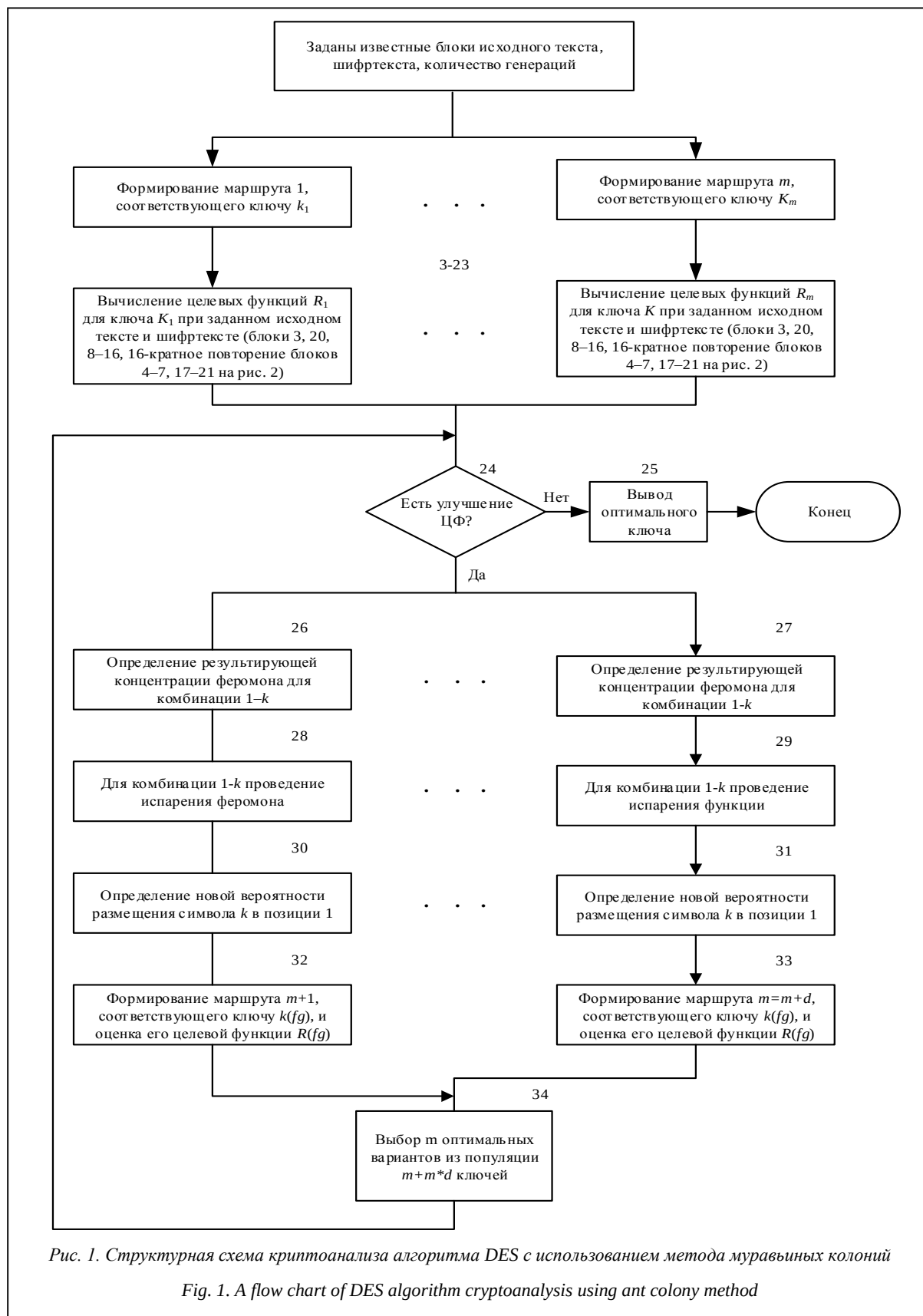
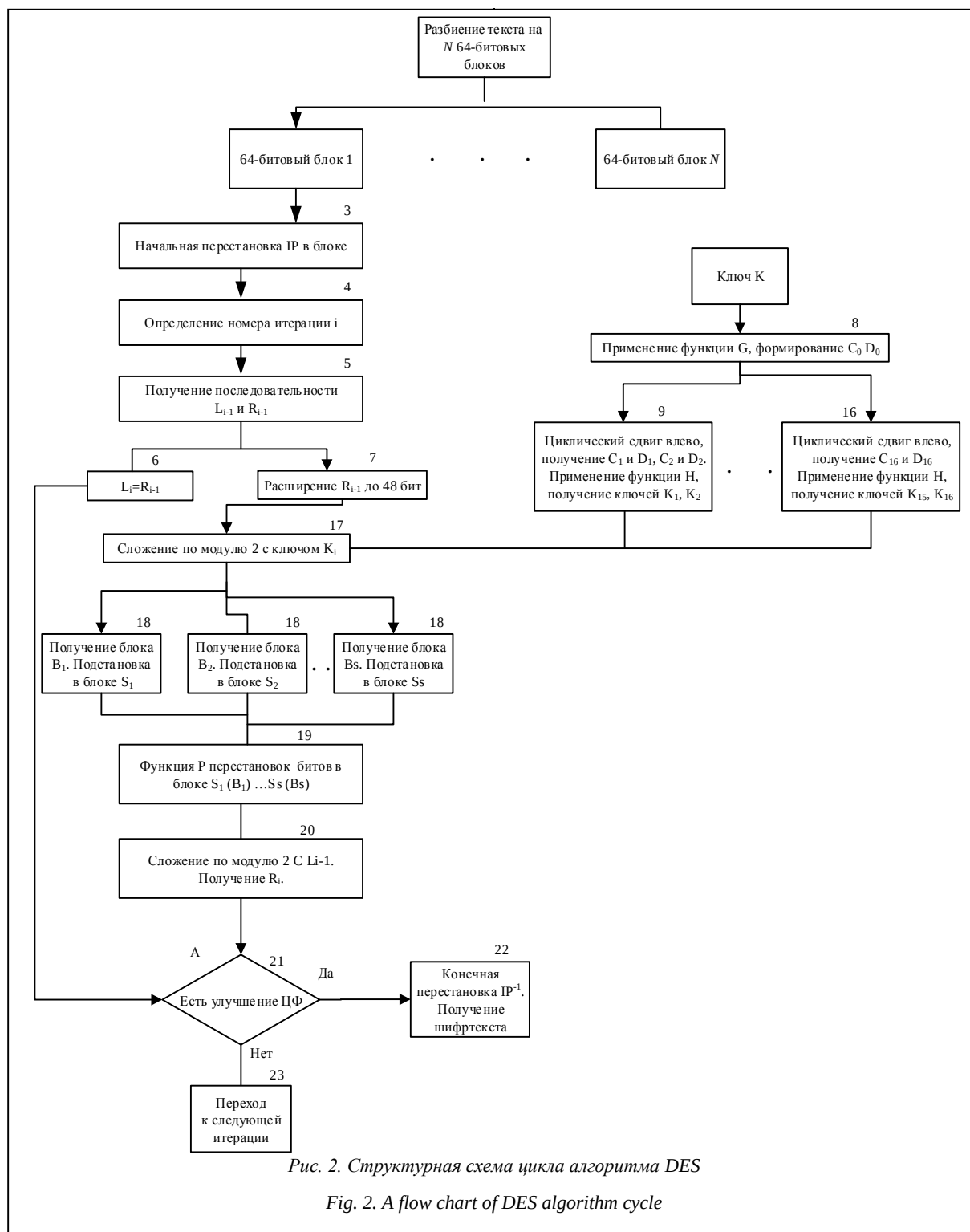


Рис. 1. Структурная схема криптоанализа алгоритма DES с использованием метода муравьиных колоний

Fig. 1. A flow chart of DES algorithm cryptanalysis using ant colony method



Используя алгоритм нахождения транзитивных связей [20, 21] и вводя для элементов l и $*$ матрицы S обозначения 1, получим матрицу S_T (матрицу следования, дополненную транзитивными связями), представленную по адресу <http://www.swsys.ru/uploaded/image/2015-4-dop/1.jpg>.

Далее введем в рассмотрение матрицу L логической несовместимости операторов, используя алгоритм, составленный для треугольной матрицы S и

описанный в [21]. Данная матрица будет содержать следующие ненулевые элементы:

$L(9, 10) = L(9, 11) = L(9, 12) = L(9, 13) = L(9, 14) = L(9, 15) = L(9, 16) = L(10, 9) = L(10, 11) = L(10, 12) = L(10, 13) = L(10, 14) = L(10, 15) = L(10, 16) = L(11, 9) = L(11, 10) = L(11, 12) = L(11, 13) = L(11, 14) = L(11, 15) = L(11, 16) = L(12, 9) = L(12, 10) = L(12, 11) = L(12, 13) = L(12, 14) = L(12, 15) = L(12, 16) = L(13, 9) = L(13, 10) = L(13, 11) = L(13, 12) =$

$L(13, 14) = L(13, 15) = L(13, 16) = L(14, 9) = L(14, 10) = L(14, 11) = L(14, 12) = L(14, 13) = L(14, 15) = L(14, 16) = L(15, 9) = L(15, 10) = L(15, 11) = L(15, 12) = L(15, 13) = L(15, 14) = L(15, 16) = L(16, 9) = L(16, 10) = L(16, 11) = L(16, 12) = L(16, 13) = L(16, 14) = L(16, 15) = L(22, 23) = L(23, 22) = L(23, 24) = L(23, 25) = L(23, 26) = L(23, 27) = L(23, 28) = L(23, 29) = L(23, 30) = L(23, 31) = L(23, 32) = L(23, 33) = L(23, 34) = L(23, 35) = L(24, 23) = L(25, 23) = L(25, 26) = L(25, 27) = L(25, 28) = L(25, 29) = L(25, 30) = L(25, 31) = L(25, 32) = L(25, 33) = L(25, 34) = L(25, 35) = L(26, 23) = L(26, 25) = L(27, 23) = L(27, 25) = L(28, 23) = L(28, 25) = L(29, 23) = L(29, 25) = L(30, 23) = L(30, 25) = L(31, 23) = L(31, 25) = L(32, 23) = L(32, 25) = L(33, 23) = L(33, 25) = L(34, 23) = L(34, 25) = L(35, 23) = L(35, 25) = 1.$

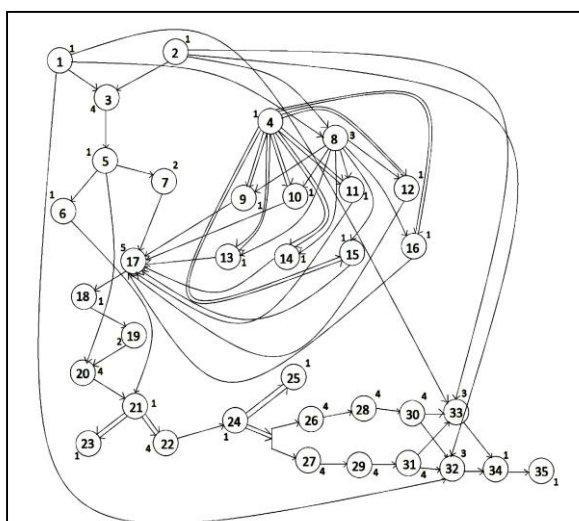


Рис. 3. Информационно-логическая граф-схема алгоритма криптоанализа с использованием метода муравьиных колоний

Fig. 3. An data-logical flow graph of a cryptoanalysis algorithm using ant colony method

Откажемся от свойства ориентированности граф-схемы и в матрице следования S_T положим $(i, j) = (j, i) = 1$. На полученную матрицу следования S' наложим матрицу логической несовместимости L , дополненную транзитивными связями. Получим матрицу независимости M , по нулевым элементам которой можно указать множество операторов, каждый из которых может быть выполнен одновременно с оператором, соответствующим номеру строки. Данная матрица будет содержать следующие элементы, равные нулю:

$M(1, 1) = M(1, 2) = M(1, 4) = M(2, 1) = M(2, 2) = M(2, 4) = M(3, 3) = M(3, 4) = M(3, 8) = M(3, 9) = M(3, 10) = M(3, 11) = M(3, 12) = M(3, 13) = M(3, 14) = M(3, 15) = M(3, 16) = M(4, 1) = M(4, 2) = M(4, 3) = M(4, 4) = M(4, 5) = M(4, 6) = M(4, 7) = M(4, 8) = M(5, 4) = M(5, 5) = M(5, 8) = M(5, 9) = M(5, 10) = M(5, 11) = M(5, 12) = M(5, 13) = M(5, 14) = M(5, 15)$

$= M(5, 16) = M(6, 4) = M(6, 6) = M(6, 7) = M(6, 8) = M(6, 9) = M(6, 10) = M(6, 11) = M(6, 12) = M(6, 13) = M(6, 14) = M(6, 15) = M(6, 16) = M(6, 17) = M(6, 18) = M(6, 19) = M(6, 20) = M(7, 4) = M(7, 6) = M(7, 7) = M(7, 8) = M(7, 9) = M(7, 10) = M(7, 11) = M(7, 12) = M(7, 13) = M(7, 14) = M(7, 15) = M(7, 16) = M(8, 3) = M(8, 4) = M(8, 5) = M(8, 6) = M(8, 7) = M(8, 8) = M(9, 3) = M(9, 5) = M(9, 6) = M(9, 7) = M(9, 9) = M(10, 3) = M(10, 5) = M(10, 6) = M(10, 7) = M(10, 10) = M(11, 3) = M(11, 5) = M(11, 6) = M(11, 7) = M(11, 11) = M(12, 3) = M(12, 5) = M(12, 6) = M(12, 7) = M(12, 12) = M(13, 3) = M(13, 5) = M(13, 6) = M(13, 7) = M(13, 13) = M(14, 3) = M(14, 5) = M(14, 6) = M(14, 7) = M(14, 14) = M(15, 3) = M(15, 5) = M(15, 6) = M(15, 7) = M(15, 15) = M(16, 3) = M(16, 5) = M(16, 6) = M(16, 7) = M(16, 16) = M(17, 6) = M(17, 17) = M(18, 6) = M(18, 18) = M(19, 6) = M(19, 19) = M(20, 6) = M(20, 20) = M(21, 21) = M(22, 22) = M(23, 23) = M(24, 24) = M(25, 25) = M(26, 26) = M(26, 27) = M(26, 29) = M(26, 31) = M(27, 26) = M(27, 27) = M(27, 28) = M(27, 30) = M(28, 27) = M(28, 28) = M(28, 29) = M(28, 31) = M(29, 26) = M(29, 28) = M(29, 29) = M(29, 30) = M(30, 27) = M(30, 29) = M(30, 30) = M(30, 31) = M(31, 26) = M(31, 28) = M(31, 30) = M(31, 31) = M(32, 32) = M(32, 33) = M(33, 32) = M(33, 33) = M(34, 34) = M(35, 35) = 0.$

Таким образом, на основе данной матрицы M можно определить семейство всех внутренне устойчивых множеств. Максимальное внутренне устойчивое множество даст оценку максимального числа процессоров, требуемых для реализации алгоритма [20]. По матрице M можно определить, что число внутренней устойчивости $\lambda = 4$.

Отсюда с учетом принятых ранее допущений следуют утверждения.

1. При реализации реального алгоритма криптоанализа, использующего множество решений, содержащее n ключей, требуемое число процессоров (P) составит n .

2. Поскольку число процессоров для оценки функции R пригодности ключей составляет 4 (блоки 3–23), при параллельной реализации алгоритмов оценки n ключей число процессоров составит не более $4 \cdot n$.

3. Аналогично при параллельной обработке результирующих концентраций феромона, его испарения и определения вероятности размещения l символов в k позиций число процессоров составит не более $l \cdot k$.

4. При параллельном формировании $m \cdot d$ ключей, образующих расширенную популяцию решений и оценки их целевой функции R , число процессоров составит не более $n = 4 \cdot m \cdot d$.

Очевидно, что при параллельной реализации алгоритма криптоанализа необходимое число процессоров не превысит максимального из отмеченных в пунктах 1–4 величин, то есть $P = \max(4 \cdot n, l \cdot k, 4 \cdot m \cdot d)$.

Для определения точной оценки числа процессоров, необходимых для реализации алгоритма при заданных временных ограничениях, воспользуемся методами, описанными в [20, 21]. Рассмотрим следующую задачу: для алгоритма криптоанализа на основе построенного информационно-логического графа и для заданного времени $T_{\text{зад}}$ найти необходимое наименьшее число процессоров однородной вычислительной системы и план выполнения на них операторов.

Данная задача рассматривалась применительно к информационно-логической схеме криптоанализа алгоритма DES с помощью генетического алгоритма в [22]. Аналогичным образом рассмотрим метод решения задачи на примере алгоритма криптоанализа стандарта DES с помощью метода муравьиных колоний, представленного информационно-логическим графом на рисунке 3. На первоначальном этапе определим скалярные веса вершин, отражающих время выполнения операций. Для определения данного времени воспользуемся основными правилами анализа программ, описанными в [23], которые определяют время выполнения операторов присваивания, чтения записи ($O(1)$), выполнения последовательности операторов (правило сумм), условных операторов, цикла.

Время выполнения операторов присваивания и элементарных логических операций примем равным $O(1)$ (блоки 6, 4, 5, 21, 23, 24, 25, 35), время перестановки с помощью матриц IP и IP^{-1} – $O(32)$ (блоки 3, 22), время перестановки с помощью матрицы P – $O(16)$ (блок 19), время выполнения функции G – $O(28)$ (блок 8), время операции циклического сдвига влево на 1 и 2 бита примем равным $O(2)$ (блоки 9–16), время выполнения функции H и побитового сложения по модулю 2 R_{i-1} с ключом K_i примем равным $O(48)$ (блок 17), время выполнения функции расширения E – $O(16)$ (блок 7), время параллельного поиска в блоках $S_1 \dots S_8$ – $O(6)$ (блок 18), время сложения 32-битового блока $S_1 \dots S_8$ с L_{i-1} примем равным $O(32)$ (блок 20).

Далее для упрощения изложения аналогично [22] примем трудоемкость формирования ключей равной $O(6)$ (блоки 1, 2), трудоемкость определения результирующей концентрации феромона, его испарения, определения вероятности размещения символов в позиции – $O(36)$ (блоки 26–31), трудоемкость формирования маршрутов, составляющих расширенное множество решений, и оценку их целевой функции – $O(28)$ (с учетом длины критического пути $T_{\text{кр}}=24$ в информационно-логической схеме алгоритма DES) (блоки 32, 33), трудоемкость выбора оптимальных вариантов из расширенной популяции – $O(6)$ (блок 34).

На рисунке 3 веса вершин указаны рядом с их номерами, промасштабированы путем деления на 10 и округлены до ближайшего целого сверху.

По данному графу определим критический путь максимальной длины $T_{\text{кр}}=43$, проходящий, напри-

мер, через вершины 1-3-5-7-17-18-19-20-21-22-24-26-28-30-33-34-35. При решении задачи определения минимального числа процессоров примем заданное время $T_{\text{зад}}=T_{\text{кр}}$.

Далее по информационно-логическому графу со скалярными весами вершин определим ранние (τ_{pi}) и поздние (τ_{pi}) сроки окончания выполнения операторов с помощью алгоритмов, описанных в [21].

Эти значения следующие:

$\tau_{p1}=1, \tau_{p2}=1, \tau_{p3}=5, \tau_{p4}=1, \tau_{p5}=6, \tau_{p6}=7, \tau_{p7}=8, \tau_{p8}=4, \tau_{p9}=\tau_{p10}=\tau_{p11}=\tau_{p12}=\tau_{p13}=\tau_{p14}=\tau_{p15}=\tau_{p16}=5, \tau_{p17}=13, \tau_{p18}=14, \tau_{p19}=16, \tau_{p20}=20, \tau_{p21}=21, \tau_{p22}=25, \tau_{p23}=22, \tau_{p24}=26, \tau_{p25}=27, \tau_{p26}=\tau_{p27}=30, \tau_{p28}=\tau_{p29}=34, \tau_{p30}=\tau_{p31}=38, \tau_{p32}=\tau_{p33}=41, \tau_{p34}=42, \tau_{p35}=43; \tau_{n35}=43, \tau_{n34}=42, \tau_{n33}=41, \tau_{n32}=41, \tau_{n31}=\tau_{n30}=38, \tau_{n29}=\tau_{n28}=34, \tau_{n27}=\tau_{n26}=30, \tau_{n25}=43, \tau_{n24}=26, \tau_{n23}=43, \tau_{n22}=25, \tau_{n21}=21, \tau_{n20}=20, \tau_{n19}=16, \tau_{n18}=14, \tau_{n17}=13, \tau_{n16}=\tau_{n15}=\tau_{n14}=\tau_{n13}=\tau_{n12}=\tau_{n11}=\tau_{n10}=\tau_{n9}=8, \tau_{n8}=7, \tau_{n7}=8, \tau_{n6}=20, \tau_{n5}=6, \tau_{n4}=7, \tau_{n3}=5, \tau_{n2}=\tau_{n1}=11.$

Используя значения τ_{pi} и τ_{pi} и методику, описанную в [21], можно найти оценку минимального числа процессоров, необходимых для выполнения операторов, входящих в каждое множество *взаимно независимых операторов* (ВНО) матрицы независимости M . Выпишем все множества ВНО, определяемые по матрице M :

{1, 2, 4}, {3, 4, 8}, {3, 9}, {3, 10}, {3, 11}, {3, 12}, {3, 13}, {3, 14}, {3, 15}, {3, 16}, {4, 5, 8}, {4, 6, 7, 8}, {5, 9}, {5, 10}, {5, 11}, {5, 12}, {5, 13}, {5, 14}, {5, 15}, {5, 16}, {6, 9, 7}, {6, 10, 7}, {6, 11, 7}, {6, 12, 7}, {6, 13, 7}, {6, 14, 7}, {6, 15, 7}, {6, 16, 7}, {6, 17}, {6, 18}, {6, 19}, {6, 20}, {26, 27}, {27, 28}, {28, 29}, {29, 26}, {29, 30}, {30, 27}, {30, 31}, {31, 26}, {31, 28}.

Построим диаграммы ранних и поздних сроков окончания выполнения операторов и произведем такое распределение временных границ операторов, при котором число используемых процессоров минимально. Рассмотрим максимальное множество ВНО {4, 6, 7, 8}. Диаграммы ранних и поздних сроков выполнения операторов имеют вид, показанный на рисунке 4.

Таким образом, возможно такое распределение выполнения операторов данного множества ВНО, при котором число процессоров равно 1.

Рассмотрим множества ВНО, содержащие 3 элемента. Для множества {3, 4, 8} диаграммы ранних и поздних сроков выполнения показаны на рисунке 5.

Таким образом, для данного множества ВНО минимальное число процессоров составит 2. Для множества ВНО {4, 5, 8} диаграммы будут иметь вид, показанный на рисунке 6.

Для данного множества ВНО требуемое число процессоров составит 1.

Рассмотрим множество ВНО {1, 2, 4} (рис. 7).

Для данного множества ВНО оценка минимального числа процессоров составит 2.

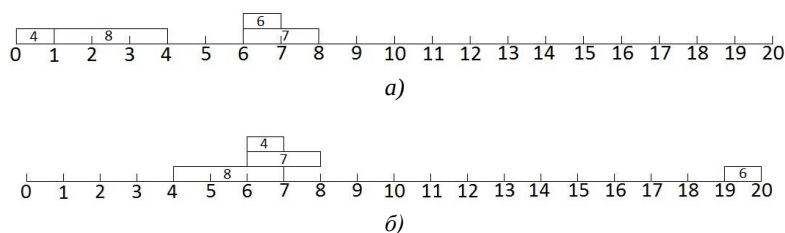


Рис. 4. Диаграммы ранних (а) и поздних (б) сроков окончания выполнения операторов для множества ВНО {4, 6, 7, 8}

Fig. 4. Diagrams of early (a) and late (б) end dates of executing statements for BHO set {4, 6, 7, 8}

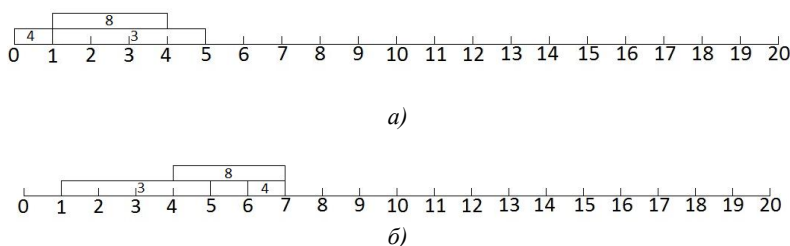


Рис. 5. Диаграммы ранних (а) и поздних (б) сроков окончания выполнения операторов для множества ВНО {3, 4, 8}

Fig. 5. Diagrams of early (a) and late (б) end dates of executing statements for BHO set {3, 4, 8}

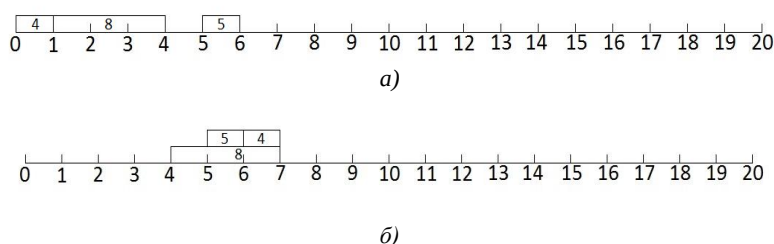


Рис. 6. Диаграммы ранних (а) и поздних (б) сроков окончания выполнения операторов для множества ВНО {4, 5, 8}

Fig. 6. Diagrams of early (a) and late (б) end dates of executing statements for BHO set {4, 5, 8}

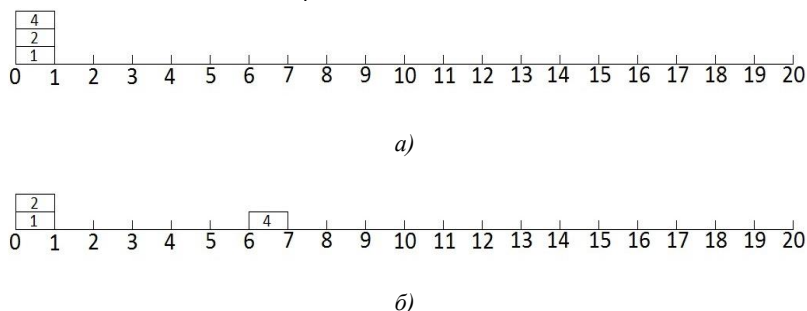


Рис. 7. Диаграммы ранних (а) и поздних (б) сроков окончания выполнения операторов для множества ВНО {1, 2, 4}

Fig. 7. Diagrams of early (а) and поздних (б) сроков окончания выполнения операторов для множества ВНО {1, 2, 4}

Поскольку остальные множества ВНО содержат операторы с равными ранними и поздними сроками окончания, рассмотрим диаграммы для одного из них, например {6, 9, 7} (рис. 8).

Для данного множества ВНО требуемое число процессоров также составит 1.

Таким образом, поскольку оценка необходимого минимального числа процессоров составляет 2, дальнейшее исследование множеств ВНО, содержащих 2 элемента, не имеет смысла.

Докажем, что эта оценка является минимальной, и определим план выполнения операторов с помощью алгоритма, представленного в [21]. Сущность данного алгоритма заключается в нахождении максимального множества взаимно независимых операторов в матрице независимости, последовательном проведении фиктивных связей в информационно-логическом графе, не увеличивающих длину критического пути. В матрице независимости M определим произвольное множество ВНО, содержащее r элементов, для которого $r > P$. Пусть выбрано множество ВНО {3, 4, 8}. Между вершинами этого множества проведем $r-P=1$ фиктивных связей в информационно-логическом графе так, чтобы выполнялось соотношение $T_{кр} \leq T_{зad}$. Связи $4 \rightarrow 3$ и $8 \rightarrow 4$ увеличивают $T_{кр}$, связь $3 \rightarrow 4$ не изменяет его. Отразим данную связь в матрице следования S и сформируем новую матрицу независимости M' .

В матрице M' определим следующее множество ВНО, содержащее более 2 операторов. Выберем множество {4, 5, 8}. Между вершинами этого множества проведем фиктивную связь

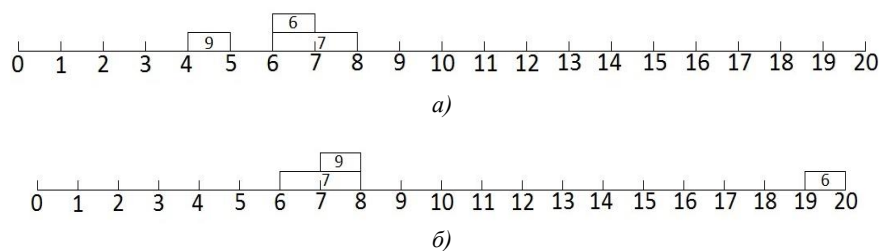


Рис. 8. Диаграммы ранних (а) и поздних (б) сроков окончания выполнения операторов для множества ВНО {6, 9, 7}

Fig. 8. Diagrams of early (a) and late (b) end dates of executing statements for BHO set {6, 9, 7}

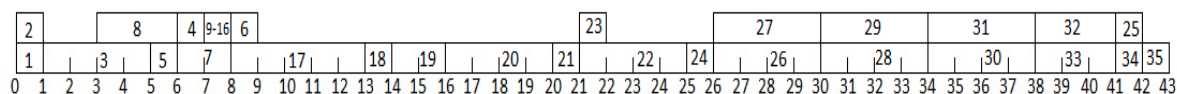


Рис. 9. План выполнения операторов муравьиного алгоритма криптоанализа за время $T_{кр}=43$ при использовании 2 процессоров

Fig. 9. An executing statements plan of ant cryptoanalysis algorithm in a time $T_{кр}=43$ when using 2 processors

5→4, не приводящую к увеличению $T_{кр}$. Получим матрицу независимости M'' .

В матрице M'' определим следующее множество ВНО, содержащее $r > P$ операторов. Это множество {4, 6, 7, 8}. Между вершинами этого множества проведем $r - P = 2$ фиктивных связей, не приводящих к увеличению $T_{кр}$. Допустимая комбинация связей 8→4→6. После ее проведения получим матрицу независимости M''' .

В матрице M''' найдем множество ВНО {6, 7, 9}. Проведем допустимую фиктивную связь 7→6. После проведения данной связи матрица независимости M'''' будет иметь вид, представленный в таблице (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/2.jpg>).

Поскольку матрица M'''' не содержит полных множеств ВНО с числом элементов, большим 2, $P=2$ – решение задачи.

Таким образом, на основе фиктивных связей, проведенных в информационно-логическом графе и определяющих порядок выполнения операторов, можно представить план выполнения операторов при заданном времени $T_{кр}=43$, показанный на рисунке 9.

Отсюда следует, что сформулированные выше утверждения 1–4 можно обобщить следующим образом.

При реализации реального алгоритма криптоанализа необходимое число процессоров составит $\max(2*m, l*k, 2*m*d)$, где m – число используемых ключей; l и k – l символов, размещенных в k позициях; $m*d$ – число ключей, составляющих расширенное множество решений.

Таким образом, рассмотрен подход к реализации криптоанализа путем применения параллельной версии алгоритма муравьиных колоний на основе построения матрицы независимости и определения множеств ВНО. Приведена оценка необходимого минимального числа процессоров при минимальном времени реализации алгоритма. Представленный в работе алгоритм наиболее эффективен при реализации криптоанализа 2-го типа, то есть при определении ключа на основе известных однозначно определенных блоков исходного текста и шифртекста [13]. При реализации криптоанализа 1-го типа в случае неоднозначного получения блоков оптимального текста из заданного блока шифртекста также возможно (как и в случае генетического алгоритма) получение некоторого множества ключей, соответствующих оптимальным вариантам исходного текста. Для каждого варианта ключа в этом случае требуется проверить, является ли он оптимальным для криптоанализа всего текста.

Литература

1. Родзин С.И. Интеллектуальные системы. О некоторых алгоритмах, инспирированных природными системами: коллектив. монография; [под ред. В.М. Курейчика]. М.: Физматлит, 2009. С. 34–45.
2. Курейчик В.В., Курейчик В.М., Родзин С.И. Концепция природных вычислений, инспирированных природными системами // Изв. ЮФУ. 2009. № 4. С. 16–24.
3. Чернышев Ю.О., Сергеев Е.О., Дубров Е.О., Крупенин А.В., Третьяков О.П. Криптографические методы и генетические алгоритмы решения задач криптоанализа: монография. Краснодар: Изд-во ФВАС, 2013. 138 с.
4. Сергеев А.С., Крупенин А.В., Третьяков О.П., Василь-

ев А.Е., Чернышев Ю.О. Биоинспирированные алгоритмы криптоанализа асимметричных алгоритмов шифрования на основе факторизации составных чисел // Информационная безопасность – актуальная проблема современности. Совершенствование образовательных технологий подготовки специалистов в области информационной безопасности: сб. тр. Краснодар: Изд-во Военной академии связи, 2011. С. 41–47.

5. Сергеев А.С. Исследование и разработка методов генетического поиска для организации криптоанализа блочных криптосистем в системах управления безопасностью и защиты информации на примере стандарта шифрования DES // Третья Междунар. конф. по проблемам управления: пленар. докл. и избран. тр. М.: Изд-во ИПУ, 2006. С. 328–335.

6. Сергеев А.С. Применение методов генетического поиска для организации криптоанализа блочных криптосистем на примере стандарта DES // Науч. мысль Кавказа (приложение). 2006. № 15. С. 185–193.

7. Сергеев А.С., Третьяков О.П., Васильев А.Е., Чернышев Ю.О. Биоинспирированные методы криптоанализа асимметричных алгоритмов шифрования на основе факторизации составных чисел // Вестн. ДГТУ. 2011. Т. 11. № 9 (60). С. 1544–1554.

8. Чернышев Ю.О., Сергеев А.С., Венцов Н.Н. Исследование и разработка методов генетического поиска для реализации криптоанализа алгоритма IDEA и решения основных теоретико-числовых задач криптографии // Вестн. РГУПС. 2009. № 3 (35). С. 70–79.

9. Сергеев А.С. Исследование и разработка информационных технологий криптоанализа в системах защиты информации на основе генетического поиска и эволюционной оптимизации на примере стандартов шифрования России // Фундаментальные прикладные проблемы приборостроения, информации и экономики: тр. 10 Междунар. науч.-практич. конф. М.: Изд-во МГУПИ, 2007. Кн. 2. С. 161–167.

10. Сергеев А.С. Разработка методов криптоанализа на основе генетического поиска при реализации стратегий и технологий информационной защиты на примере стандартов шифрования России // Коммуникативные стратегии информационного общества: тр. Междунар. науч.-технич. конф. СПб: Изд-во политех. ун-та, 2007. С. 56–65.

11. Лебедев О.Б. Трассировка в канале методом муравьи-

ной колонии // Изв. ЮФУ. Сер. Интеллектуальные САПР. 2009. № 4. С. 46–52.

12. Муравьиные алгоритмы. URL: <http://rain.ifmo.ru/cat/data/theory/unsorted/ant-algo-2006/article.pdf> (дата обращения: 15.05.2015).

13. Чернышев Ю.О., Сергеев А.С., Дубров Е.О., Рязанов А.Н. Применение метода муравьиных колоний для реализации криптоанализа блочных криптосистем // Программные продукты и системы. 2014. № 1 (105). С. 10–19.

14. Куликов А.Л., Петрова И.Ю. Выбор оптимального блочного алгоритма шифрования трафика в информационной системе вуза. URL: <http://systech.miem.edu.ru/2004/n2/Kulikov2.htm> (дата обращения: 15.05.2015).

15. Бабенко Л.К., Ищукова Е.А. Современные алгоритмы блочного шифрования и методы их анализа. М.: Гелиос АРВ, 2006. 376 с.

16. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. М.: Радио и связь, 2001. 376 с.

17. Шнайер Б. Прикладная криптография. М.: Триумф, 2002. 816 с.

18. Тимченко С.В. Сравнение трех подходов к построению параллельных генетических алгоритмов на примере некоторых задач функциональной оптимизации и генетического программирования. URL: <http://www.botik.ru/PSI/RCMS/publications/publ-texts/2005/grishagine.pdf> (дата обращения: 15.05.2015).

19. Дискретная математика: алгоритмы. URL: <http://rain.ifmo.ru/cat/view.php/theory/unsorted/genetic-2005> (дата обращения: 15.05.2015).

20. Барский А.Б. Планирование параллельных вычислительных процессов. М.: Машиностроение, 1980. 191 с.

21. Сергеев А.С. Параллельное программирование. Р-н-Д: Издат. центр ДГТУ, 2002. 77 с.

22. Сергеев А.С. Разработка генетического метода криптоанализа блочных криптосистем и исследование возможности их параллельной реализации в системах защиты информации на примере стандарта DES // Системный анализ в проектировании и управлении: тр. 10 Междунар. науч.-практич. конф. СПб: Изд-во политех. ун-та, 2006. С. 258–265.

23. Ахо Альфред В., Джон Ульман Джеффри Д. Структуры данных и алгоритмы. М.: Вильямс, 2003. 384 с.

DOI: 10.15827/0236-235X.112.148-157

Received 24.06.15

DEVELOPMENT AND RESEARCH OF A PARALLEL ANT COLONY ALGORITHM FOR A BLOCK CRYPTOSYSTEM CRYPTANALYSIS

Chernyshev Yu.O., Dr.Sc. (Engineering), Professor, myvnn@list.ru;

Sergeev A.S., Ph.D. (Engineering), Doctoral Student, sergeev00765@mail.ru;

Ryazanov A.N., Postgraduate Student, alexandr_r89@mail.ru

(Don State Technical University, Gagarin Sq. 1, Rostov-on-Don, 344000, Russian Federation);

Kapustin S.A., Ph.D. (Engineering), Associate Professor, skappa@yandex.ru

(Military Academy of Communications, Krasnodar Branch, Krasina St. 4, Krasnodar, 350035, Russian Federation)

Abstract. The article considers the possibility of parallel implementation of ant colonies algorithms for block cryptosystems cryptanalysis. The authors specify that a new scientific direction “natural calculations” is relevant, provide the block diagram of DES standard cryptanalysis using a method of ant colonies. The parallel version of cryptanalysis algorithm is described on the basis of a data-logical flow graph, sequence matrices, logical incompatibility and independence. The minimum number of the processors for a cryptanalysis algorithm is defined on the basis of a technique of defining the number of processors. This technique includes finding of a mutually independent operators maximal set in an independence matrix, consecutive fictive linking in a data-logical graph provided that these links don't increase a critical way length. A bioinspired cryptanalysis method application is distinguished by the possibility of using an encryption (decryption) algorithm as an objective function of key assessment defined by genetic operations. Therefore, when using the bioinspired cryptanalysis methods the process of private key definition (for example, at type 2 cryptanalysis) depends not on encryption complexity, but on the bioinspired method itself that provides a sufficient variety of key generation. This proves the relevance of the problem connected with a research on the bioinspired algorithms application possibility for block cryptosystems cryptanalysis.

Keywords: cryptanalysis, ant algorithm, matrix of logical incompatibility, matrix of independence, mutually independent operators maximal set.

References

1. Rodzin S.I. *Intellektualnye sistemy. O nekotorykh algoritmakh, inspirirovannykh prirodnymi sistemami* [Intelligent Systems. On Some Algorithms that are Inspired by Natural Systems]. Collective monograph. V.M. Kureychik (Ed.). Moscow, Fizmatlit Publ., 2009, pp. 34–45.
2. Kureychik V.V., Kureychik V.M., Rodzin S.I. Concept evolutionary computation is inspired by natural systems. *Izvestiya YuFU* [Izvestiya SFedU. Engineering Sciences]. 2009, no. 4, pp. 16–24 (in Russ.).
3. Chernyshev Yu.O., Sergeev E.O., Dubrov E.O., Krupenin A.V., Tretyakov O.P. *Kriptograficheskie metody i geneticheskie algoritmy resheniya zadach kriptanaliza* [Cryptographical Methods and Genetic Algorithms for Cryptanalysis Problems]. Monograph. Krasnodar, 2013, 138 p.
4. Sergeev A.S., Krupenin A.V., Tretyakov O.P., Vasilyev A.E., Chernyshev Yu.O. Cryptanalysis bioinspired algorithms for asymmetric encryption algorithms based on composite numbers factorization. *Sb. trudov "Informatsionnaya bezopasnost – aktualnaya problema sovremennosti. Sovershenstvovanie obrazovatelnykh tekhnology podgotovki spetsialistov v oblasti informatsionnoy bezopasnosti"* [Proc. "Information security is an important problem of modern times. Improvement of specialist training technologies in information security"]. Krasnodar, 2011, pp. 41–47 (in Russ.).
5. Sergeev A.S. Research and development of genetic search to organize block cryptosystems cryptanalysis in security management system in case of DES encryption. *Materialy 3 Mezhdunar. konf. po problemam upravleniya: plenarnye doklady i izbrannye trudy* [Proc. of the 3rd Int. Conf. on Management Problems]. Moscow, Institute of Control Sciences Publ., 2006, pp. 328–335 (in Russ.).
6. Sergeev A.S. Using genetic search methods to organize block cryptosystems cryptanalysis in case of DES encryption. *Nauchnaya mysl Kavkaza (prilozhenie)* [Scientific Thought of Caucasus (supplement)]. 2006, no. 15, pp. 185–193 (in Russ.).
7. Sergeev A.S., Tretyakov O.P., Vasilev A.E., Chernyshev Yu.O. Bioinspired cryptanalysis methods of asymmetric encryption algorithms based on composite factorization. *Vestnik DGTU* [Bulletin of DSTU]. 2011, vol. 11, no. 9 (60), pp. 1544–1554 (in Russ.).
8. Chernyshev Yu.O., Sergeev A.S., Ventsov N.N. Research and development of genetic search to organize IDEA algorithm cryptanalysis and solve main number-theoretic cryptography problems. *Vestnik RGUPS* [Bulletin of RSTU]. 2009, no. 3 (35), pp. 70–79 (in Russ.).
9. Sergeev A.S. Research and development of cryptanalysis IT in information security systems based on genetic search and evolution optimization in case of Russian encryption standards. *Nauch. tr. Yubileynoy 10 Mezhdunar. nauchno-praktich. konf. "Fundamentalnye prikladnye problemy priborostroeniya, informatsii i ekonomiki"* [Proc. of the 10th Anniv. Int. Science and Practice Conf. "Fundamental Applied Problems of Instrumentation, Information and Economics"]. Moscow, 2007, vol. 2, pp. 161–167 (in Russ.).
10. Sergeev A.S. Development of cryptanalysis methods based on genetic search when implementing information security strategies and technologies in case of Russian encryption standards. *Trudy mezhdunar. nauchno-tekhnich. konf. "Kommunikativnye strategii informatsionnogo obshchestva"* [Proc. of the Int. Science and Tech. Conf. "Communicative strategies of information society"]. St.-Petersburg, 2007, pp. 56–65 (in Russ.).
11. Lebedev O.B. Chanel routing bases on method of ant colony optimization. *Izvestiya YuFU. Tekhnicheskie nauki* [Izvestiya SFedU. Engineering Sciences]. 2009, no. 4, pp. 46–52 (in Russ.).
12. Muravyinye algoritmy [Ant algorithms]. Available at: <http://rain.ifmo.ru/cat/data/theory/unordered/ant-algo-2006/article.pdf> (accepted May 15, 2015).
13. Chernyshev Yu.O., Sergeev A.S., Dubrov E.O., Ryazanov A.N. Ant colony method application to implement cryptanalysis of block cryptosystems. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 1 (105), pp. 10–19 (in Russ.).
14. Kulikov A.L., Petrova I.Yu. *Vybor optimalnogo blochnogo algoritma shifrovaniya trafika v informatsionnoy sisteme vuza* [Choosing Optimal Block Algorithm for Traffic Encryption in a University Information System]. Available at: <http://systech.miem.edu.ru/2004/n2/Kulikov2.htm> (accepted May 15, 2015).
15. Babenko L.K., Ishchukova E.A. *Sovremennyye algoritmy blochnogo shifrovaniya i metody ikh analiza* [Modern Block Encryption Algorithms and their Analysis]. Moscow, Gelios ARV Publ., 2006, 376 p.
16. Romanets Yu.V., Timofeev P.A., Shangin V.F. *Zashchita informatsii v kompyuternykh sistemakh i setyakh* [Information Security in Computer Systems and Networks]. Moscow, 2001, 376 p.
17. Shnayer B. *Prikladnaya kriptografiya* [Applied cryptography]. Moscow, 2002, 816 p.
18. Timchenko S.V. *Sravnienie trekh podkhodov k postroeniyu parallelnykh geneticheskikh algoritmov na primere nekotorykh zadach funktsionalnoy optimizatsii i geneticheskogo programmirovaniya* [Comparing 3 Approaches to Building Parallel Genetic Algorithms in Case of Some Functional Optimization Problems and Genetic Programming]. Available at: <http://www.botik.ru/PSI/RCMS/publications/publ-texts/2005/grishagine.pdf> (accepted May 15, 2015).
19. *Diskretnaya matematika: algoritmy* [Discrete Mathematics: Algorithms]. Available at: <http://rain.ifmo.ru/cat/view.php/theory/unordered/genetic-2005> (accepted May 15, 2015).
20. Barsky A.B. *Planirovanie parallelnykh vychislitelnykh protsessov* [Planning Parallel Computing Processes]. Moscow, 1980, 191 p.
21. Sergeev A.S. *Parallelnoe programmirovaniye* [Parallel Programming]. Rostov-on-Don, 2002, 77 p.
22. Sergeev A.S. Development of block cryptosystems cryptanalysis genetic method and research on the possibility of their parallel implementation in information security systems in case of DES. *Sistemnyy analiz v proektirovani i upravlenii: Trudy 10 Mezhdunar. nauch.-praktich. konf.* [System analysis in design and management: proc. of the 10th Int. Science and Practice Conf.] St.-Petersburg, 2006, pp. 258–265 (in Russ.).
23. Aho A.V., Ullman J.D., Hopcroft J.E. *Data Structures and Algorithms*. Pearson Publ., 1983, 427 p. (Russ. ed.: Moscow, 2003, 384 p.).

УДК 658.012.011:656.2.001.57+06

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.158-165

ОЦЕНКА НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ МЕТОДАМИ ДИСКРЕТНО-СОБЫТИЙНОГО МОДЕЛИРОВАНИЯ

*(Работа выполнена при финансовой поддержке РФФИ, проекты №№ 15-08-01886-а, 15-01-03067-а)**М.А. Бутакова, д.т.н., профессор, декан, butakova@rgups.ru;**А.Н. Гуда, д.т.н., профессор, проректор, guda@rgups.ru;**А.В. Чернов, д.т.н., профессор, зав. кафедрой, avche@yandex.ru;**С.В. Чубейко, к.т.н., greyc@mail.ru**(Ростовский государственный университет путей сообщения,
пл. им. Ростовского Стрелкового полка Народного ополчения, 2, г. Ростов-на-Дону, 344038, Россия)*

В статье рассматривается дискретно-событийное моделирование и представлены его отличительные особенности от других видов моделирования. Основное отличие – отсутствие привязки ко времени: достаточно соблюдать последовательность наступления событий, при этом не важно, какой временной промежуток будет между событиями. Дано определение модели дискретно-событийной системы с дополнением ее модельными часами, которые воспроизводят хронологию событий. Решается важная задача генерации списка событий различными способами: объектно-ориентированное и процессно-ориентированное исполнение событий. Подробно рассматриваются оба способа: приводятся иллюстрация, алгоритм и элемент программной реализации. События могут объединяться в группы, которые называются процессами. Процессно-ориентированное моделирование сложнее объектно-ориентированного, так как имеется планировщик процессов. Также в статье рассматривается оценка надежности ПО, базирующаяся на дискретно-событийном подходе. Данный подход основан на идее роста надежности ПО. Поиск ошибок моделируется случайным точечным процессом. При обнаружении ошибки она устраняется, тем самым ПО становится более надежным. Моделирование делится на две части: генерация процессов, имитирующих появление ошибок в ПО, и оценка системной надежности компонентного ПО. В статье рассматриваются варианты расчета вероятности возникновения ошибки в зависимости от структуры программ: последовательная, разветвляющаяся, циклическая и параллельная структура программного компонента. Для каждого варианта представлен иллюстрирующий рисунок и приведена вычислительная схема. Для циклической схемы программного компонента используется вычислительная схема последовательного компонента, так как это своего рода однотипные повторы последовательной структуры программного компонента.

Ключевые слова: дискретно-событийное моделирование, событийно-ориентированное моделирование, процессно-ориентированное моделирование, надежность программного обеспечения, структура программного компонента.

В данной работе рассматриваются аспекты моделирования широкого класса систем, причем именно признаки событийности являются наиболее существенными для адекватного составления математических моделей. Признаки событийности и построенные на них методы моделирования [1–3] существенно отличают исследуемый подход от методов моделирования, общепринятых, например, в теории систем и теории автоматического управления, основными инструментами которых являются интегро-дифференциальные уравнения, методы теории вероятностей, математической статистики и случайных процессов. В теории систем и теории автоматического управления обычным описанием исследуемой системы является описание вход-выход, а изменение выхода относительно входа, то есть пространство состояний системы, задается передаточными функциями в матричном виде. Большинство систем, являющихся объектами моделирования, нелинейные, и в данном случае существенные усилия моделирования направлены на линеаризацию систем, выполняемую путем решения систем дифференциальных уравнений в матричном виде. Результатами моделирования являются восстановленное фазовое пространство моделируемых систем и характеристики звеньев управления и регулирования.

Другим подходом является имитационное моделирование [4], использующее методы теорий систем и сетей массового обслуживания (СМО и СеМО соответственно) [5, 6], в которых рассматриваются различные модели входных, выходных потоков и правил обслуживания, построенных на базе соответствующих законов распределения случайных величин и процессов. В данном случае методами моделирования являются генераторы некоторых типов случайных процессов (имитирующих моменты времени поступления заявок на обслуживание), а результатами моделирования (часто усредненными) – времена пребывания заявки в очереди, системе, времена обслуживания, вероятности пребывания в очереди, вероятности обслуживания (за некоторый период) и другие вероятностные и статистические характеристики.

Заметим, что и в общей теории систем, и в теориях СМО и СеМО неявно предполагается использование процессного времени. Это означает, что в первом случае принимается, что процессы в системах протекают по возможности мгновенно (хотя в некоторых случаях допустима их инерционность), а во втором случае принимается, что процессы подчинены некоторому вероятностному закону. Однако во втором случае иногда рассматривают потоки *general* (общего) типа, в которых основным

соотношением является рекуррентное уравнение Линдли [7], что по сути близко к рассматриваемому далее подходу с идейной стороны, но не со стороны реализации методов моделирования. В целом процессное время означает, что изменение состояний системы, а также ее модели можно отметить на некоторой временной шкале; если шкала является непрерывной, то естественным будет непрерывное моделирование состояний системы, иначе – дискретное, а также соответствующий им математический аппарат.

В основу дискретно-событийного моделирования, развиваемого появления известной системы GPSS [8] и сетей Петри [9, 10] положена другая концепция: состояния системы изменяются под воздействием некоторых событий, в общем случае безотносительно к их точной привязке к временной шкале. Существенными являются лишь факты возникновения этих событий и взаимодействие их между собой, то есть синхронизация (некоторое событие предшествует другому, некоторое событие вызывает возникновение другого либо других событий и так далее). Примером таких информационных систем являются сетевые компьютерные системы. Для сетевых компьютерных систем как многопользовательских многозадачных, многомашинных, многопроцессорных систем характерным является еще один аспект событийности – конкуренция за сетевые распределенные вычислительные ресурсы с целью увеличения производительности, минимизации простоев и тому подобное. Оба этих аспекта (синхронизация и конкуренция) делают сетевые компьютерные системы существенно нелинейными, что усложняет их аналитическое и имитационное моделирование, а рассматриваемый далее в работе подход можно воспринимать как возможную линейризацию таких систем.

Подходы и алгоритмы дискретно-событийного моделирования

Проведем грань между дискретно-событийным моделированием и другими методами моделирования более четко. Как уже упоминалось, большинство систем моделируется по принципу «вход–состояние–выход». Принимая общеизвестные обозначения векторов: $\mathbf{u}(t)$ – вход, $\mathbf{x}(t)$ – состояние, $\mathbf{y}(t)$ – выход, динамика моделируемой системы описывается уравнениями

$$\mathbf{x}'(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \quad (1)$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \quad (2)$$

с начальными условиями $t > 0$.

Уравнение (1) означает составление множества состояний моделируемой системы, а если принять, что множество таких состояний равно n , а множество входных сигналов равно m , то получается, что необходимо моделировать n уравнений состояний:

$$x'_1(t) = f_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t),$$

...

$$x'_n(t) = f_n(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t)$$

и k выходных уравнений системы:

$$y_1(t) = g_1(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t),$$

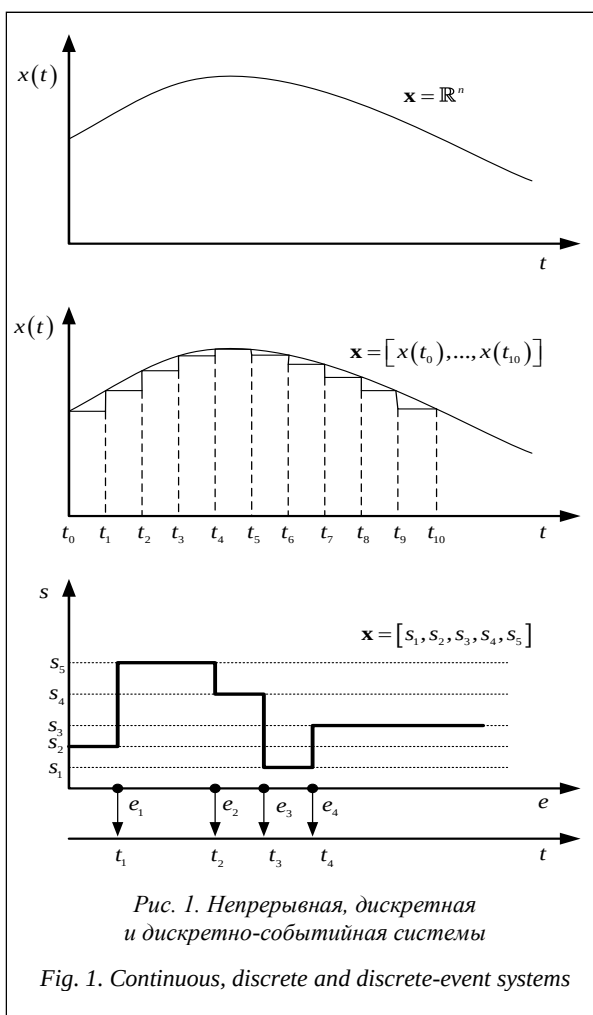
...

$$y_k(t) = g_k(x_1(t), \dots, x_n(t), u_1(t), \dots, u_m(t), t)$$

с начальными условиями $x_1(t_0) = x_1, \dots, x_n(t_0) = x_n$.

В зависимости от вида функций f и g , способа фиксации моментов времени t в (1, 2) осуществляется непрерывное (или дискретное) моделирование нелинейной (или линейной) динамической системы.

Еще раз следует заметить, что изменение состояний динамической системы в таких случаях моделирования всегда привязано ко времени, какими бы его измерениями мы ни пользовались, непрерывными или дискретными. В дискретно-событийном моделировании важен факт фиксации события (или группы событий), а в какое время, либо интервал времени, либо через какой промежуток времени эти события фиксируются, уже не столь важно. Получается, что «событие первично, а время вторично». В работе [11] имеются иллюстрации к вышесказанным положениям, которые приведены в доработанном виде на рисунке 1. В верхней части



показано пространство состояний непрерывной системы, в средней – дискретной системы, а в нижней – дискретно-событийной системы. Ясно, что пространство состояний дискретно-событийной системы является дискретным и составляет события $s_1, \dots, s_5$, а переключение между этими состояниями происходит в соответствии с наступлением некоторых событий $e_1, \dots, e_4$. Естественным, при динамической смене состояний системы может происходить возврат к предыдущим состояниям, поэтому моделирование пространства состояний будет составлять при упорядочении по хронологии цепочек событий и совпадении момента времени и события (или группы событий) последовательность пар («время», «состояние»). В данном случае (рис. 1) это $e_0 = s_2, \{e_1, e_2, e_3, e_4\} =$

$$= \{(t_1, s_5), (t_2, s_4), (t_3, s_1), (t_4, s_3)\}. \quad (3)$$

Событийное функционирование обнаруживается у широкого класса современных систем. В области информационных систем и технологий событийными являются объектно-ориентированные программные системы, сетевые компьютерные, интерактивные, диалоговые системы и другие.

В формальном виде дискретно-событийная система – это некая разновидность временного автомата [11], который представляется в следующем виде.

Определение 1. Модель дискретно-событийной системы представляет собой кортеж

$$DS = (X, E, f, \Gamma, x_0), \quad (4)$$

где X – конечное множество, пространство состояний системы; E – конечное множество событий; f – функция смены состояний, $f: X \times E \rightarrow X$; Γ – конечное множество активных (и исполняемых в текущий момент) событий; x_0 – начальное состояние.

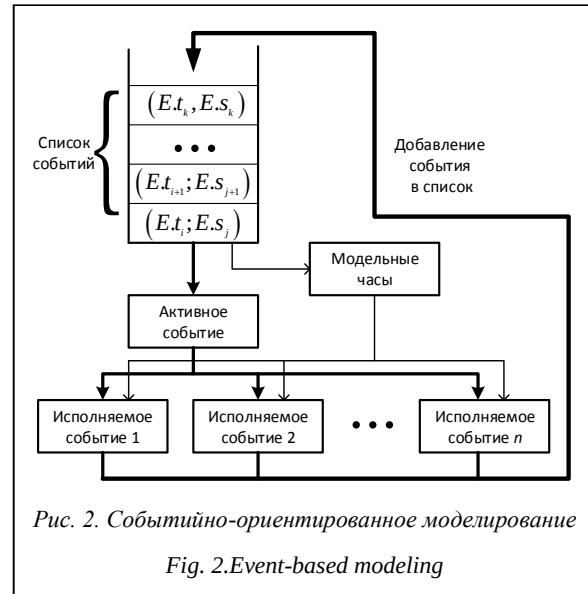
В связи с тем, что время как таковое не присутствует в модели (4), но при имитационном моделировании все же необходимо воспроизводить хронологию событий по мере упорядоченности их между собой, (4) дополняется модельными часами, связанными с множеством событий. Такие модельные часы представляют собой конечное множество

$$V = \{v_i : i \in E\}, \quad v_i = \{v_{i,1}, v_{i,2}, \dots\}, \quad (5)$$

где $v_{i,k}$ – время жизни (продолжительность) события.

Для последовательности событий $\{e_1, e_2, \dots, e_k, e_{k+1}, \dots\}$ можно «включить модельные часы» (5), получить $V = \{v_i : i = 1, \dots, m\}$ и генерировать события $e_{k+1} = h(x_k, v_1, \dots, v_m)$. Динамика состояний дискретно-событийной системы при этом определяется уравнением $x_{k+1} = f(x_k, v_1, \dots, v_m)$.

Таким образом, очень существенной задачей является формирование (генерация) списков событий, то есть пар вида (3), в зависимости от которой можно выделить событийно-ориентированное и процессно-ориентированное исполнение событий. Рассмотрим их более подробно.



Событийно-ориентированное моделирование в дискретно-событийной системе проиллюстрировано на рисунке 2.

Список событий является динамической структурой, а модельные часы содержат время последнего исполняемого события. В алгоритмическом виде моделирование заключается в следующей последовательности действий.

Алгоритм 1. Событийно-ориентированное моделирование.

1. Установить модельные часы в 0. Инициализировать начальный список событий, расположив их в хронологическом порядке следования. Элемент списка событий имеет структуру $(E.t_i; E.s_j)$ и характеризуется временем и типом состояния (t_i, s_j) .
2. Выбрать событие E из начала списка. Если список пуст, завершить моделирование.
3. Установить модельные часы в $E.t_i$. Проверить длительность события и при превышении времени, отведенного на моделирование, завершить его.
4. В соответствии с типом события и состоянием $E.s_j$ исполнить подпрограмму-обработчик события.
5. Обновить список событий, системные переменные и структуры, поместить новое событие в список событий.
6. Продолжить моделирование, перейдя к п. 2.

Элементы возможной программной реализации алгоритма 1 показаны в примере 1.

Пример 1. Основной событийно-ориентированный обработчик.

```
event_oriented()
{
    sim_time = 0; // Начальное время = 0
    list_init(); // Инициализация списка событий
    done = FALSE; // Признак завершения моделирования
    while(!done) // Основной обработчик событий
    {
        next_event(status, time); // Выбор события из списка
        stime = time; // Фиксация времени
```

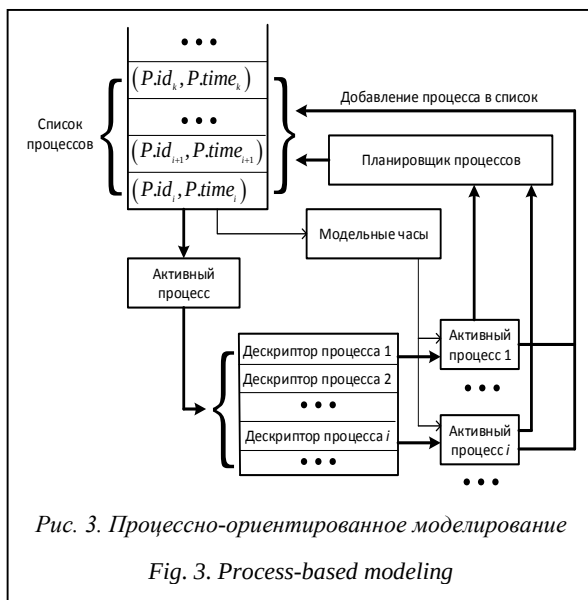
```

if (stime > max_sim_time) // Проверка превышения вре-
мени
{
    done = TRUE; // Установка признака завершения
    break; // Выход по завершении
}
exec_event(status); // Обработка события
}

```

Обработчик устанавливает начальное время, инициализирует список событий и в цикле, выбирая следующее событие, переустанавливает модельные часы, проверяя, чтобы не было превышения максимально допустимого времени моделирования, вызывает подпрограмму `exec_event(status)`. При ее реализации имеет смысл запрограммировать хотя бы простейшее планирование списка событий в виде динамической очереди FIFO (First In First Out).

Во втором подходе, процессно-ориентированном исполнении событий, есть возможность исполнения группы событий при их логическом объединении в процессы, как показано на рисунке 3.



Основным отличием данного вида от предыдущего является не только возможность объединять процессы в группы, но и планировать переключение между процессами, разделяя при этом совместные ресурсы моделируемой системы. Общая схема такого моделирования представлена в виде алгоритма 2.

Алгоритм 2. Процессно-ориентированное моделирование.

1. Установить модельные часы в 0. Инициализировать начальный список событий, расположив их в хронологическом порядке следования. Элементы списков событий группируются в процессы, имеющие структуру $(P.id; P.time_i)$, и характеризуются идентификатором и временем $(id; time)$. Запустить основной цикл моделирования.

2. Создать активный процесс P из списка планируемых к исполнению событий. Если список пустой, завершить моделирование.

3. Проверить превышение максимального времени, отведенного на моделирование. Если есть превышение, моделирование завершить.

4. Присвоить процессу дескриптор исполнения и передать его планировщику процессов.

5. Выполнить планирование процессов: исполнение, ожидание, переключение, завершение.

6. В рамках процесса требуется исполнить обработчики событий, генерировать новые события.

7. Перейти к п. 2.

В примере 2 приведена схема возможной программной реализации.

Пример 2. Схема процессно-ориентированного обработчика

```

event_job(descriptor) // Планировщик процессов (пример)
{
    while (TRUE)
    {
        wait(resource); // Ожидать свободные ресурсы
        exec_event(status, time); // Исполнять обработчик события
        switch(descriptor); // Переключать на другой процесс
        signal(release_event); // Завершить процесс
    }
}
process_init() // Инициализация процесса
{
    for (i=0; i<N; i++)
        event_job(descriptor); // Планировать процесс
    if (stime < max_sim_time)
    {
        next_event(status, time) // Выбор события из списка
        stime = time; // Фиксация времени
    }
    else break;
}
}
process_oriented() /*Основной процессно-ориентированный обработчик*/
{
    stime = 0;
    process_init();
    simulate(stime);
}

```

Возможной простейшей стратегией планировщика могут являться FCFS (First Come First Served), «первый пришел – первый обслужен», а также реализация семафорного переключения контекста процессов.

Таким образом, научные исследования в области дискретно-событийного моделирования сейчас достаточно широко развиты в различных направлениях [12–15], а также являются значительным дополнением к методам имитационного моделирования систем. Существуют также программные реализации дискретно-событийных систем моделирования, например AnyLogic, SimPy, SimEvents и другие программные системы.

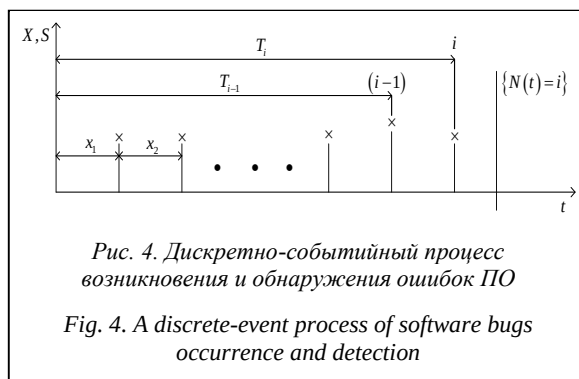
Дискретно-событийный подход к оценке надежности ПО

В целом дискретно-событийный подход опирается на упомянутые ранее идеи роста надежности ПО, однако в отличие от известных моделей будет использоваться покомпонентная технология моделирования событий возникновения ошибок на входе и выходе программного компонента. Процессы обнаружения и устранения ошибок моделируются случайными точечными процессами, а времена обнаружения ошибок сопоставляются с событиями, при возникновении которых требуется рассчитать вероятностные оценки надежности программных компонентов, которые в зависимости от реализуемых ими алгоритмов будут иметь разные формулы для расчета.

Процедуры моделирования в предлагаемом подходе можно разделить на две группы. Первая группа предназначена для генерации процессов, имитирующих появление ошибок в ПО. Сгенерированные модельные значения разделяются на несколько классов и составляют исходные данные для следующей группы процедур, которая предназначена для оценки системной надежности компонентного ПО. Модельные значения, разделенные на классы, сопоставляются с различными классами ошибок в событийной модели типа «вход–структура–выход». При этом анализируемый с точки зрения надежности программный компонент также относится к одному из классов в зависимости от используемых в нем программных конструкций. Каждому варианту программной конструкции соответствует отдельный вариант расчета оцениваемой надежности ПО.

Рассмотрим подробнее модели первой группы. Определим следующие переменные: $N(t)$ – кумулятивное (накопленное) число ошибок ПО, выявленных на стадиях разработки и тестирования до временного момента t ; T_i – i -й интервал времени возникновения либо обнаружения ошибки ($T_0=0$, $i = 1, 2, \dots$); X_i – интервал времени между $(i-1)$ -й и i -й ошибками ($X_0 = 0$, $i = 1, 2, \dots$).

На рисунке 4 показано возникновение события $\{N(t) = i\}$, которое означает, что было обнаружено i ошибок к моменту времени t .



Таким образом, имеем следующий принцип моделирования: $T_i = \sum_{k=1}^i x_k$, $x_i = T_i - T_{i-1}$.

Обозначим $S_i(x)$ уровень ошибок, то есть величину, пропорциональную числу необнаруженных (и, следовательно, неисправленных) ошибок для каждого из интервалов времени x_i , $i = 1, 2, \dots$, общий подход к дискретно-событийному моделированию можно определить как

$$S_i(x) = M_i(x) \lambda_i(x), \quad (6)$$

$$i = 1, 2, \dots, N, \quad x \geq 0, \quad \lambda(x) > 0,$$

где $M_i(x)$ – функция, задающая начальные условия модели; $\lambda(x)$ – функция интенсивности ошибок.

Естественно, в законе моделирования (6) функции $M_i(x)$ и $\lambda_i(x)$ могут быть заданы различными зависимостями – от константы до выражения, имеющего сложную форму. Сама же функция надежности ПО (обозначим ее $R_n(x)$) исходя из (6) также может иметь разный вид, например:

$$R_n(x) = \exp \left[- \int_0^x S_i(x) dx \right], \quad i = 1, 2, \dots,$$

а числовая характеристика, определяющая среднее время между ошибками, может иметь вид

$$E[X_n] = \int_0^\infty R_n(x) dx.$$

Последовательность процедур моделирования, составляющих первую группу методов моделирования, следующая.

1. Установление исходных положений моделирования.

1.1. Ошибки ПО возникают в случайные интервалы времени, отсчитываемые последовательно на одной временной оси; имеется некоторое число скрытых ошибок ПО, которые можно обнаружить на этапах разработки и тестирования.

1.2. Ошибка в ПО вызывает ошибочное состояние всей системы и требует ее исправления для восстановления функциональности системы.

1.3. Процесс исправления ошибки выполняется немедленно, новые ошибки при этом не вносятся.

2. Выбор математического аппарата моделирования.

Выбираем случайный считающий процесс $N(t)$, $t \geq 0$, подсчитывающий кумулятивное число ошибок в ПО, детектированных ко времени t .

Условия и ограничения следующие: $N(0) = 0$.

2.1. $\{N(t), t \geq 0\}$ – имеет независимые приращения.

2.2. $\Pr\{N(t + \Delta t) - N(t) = 1\} = h(t\Delta t) o(\Delta t)$, где $h(t)$ – функция мгновенной интенсивности детектирования ошибок.

2.3. $\Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t)$ – в один момент времени обнаруживается не более одной ошибки.

3. Генерация модельного процесса.

Генерируем неоднородный случайный пуассоновский процесс:

$$\Pr\{N(t) = n\} = \frac{\{H(t)\}^n}{n!} \exp[-H(t)], \quad (7)$$

$n = 0, 1, 2, \dots$, где $H(t)$ – ожидаемое кумулятивное число детектированных ошибок,

$$E[N(t) = n] = H(t) = \int_0^t h(x) dx,$$

где $h(x)$ – интенсивность обнаружения и исправления ошибок.

4. Переход к численному и алгоритмическому моделированию процессов.

В упрощенной форме выражение (7) моделируется дискретным неоднородным пуассоновским процессом со средним кумулятивным количеством детектированных ошибок D_n следующим образом:

$$\Pr\{N_x = n\} = \frac{[D_n]^x}{x!} \exp[-D_n], \quad (8)$$

($x, n = 0, 1, 2, \dots$).

В дальнейшем вместо выражения (8) могут использоваться другие различные дискретные аналоги выражения (7).

Рассмотрим один из таких аналогов, построенный на основе экспоненциальной модели роста. Пусть H_n обозначает кумулятивное число обнаруженных ошибок за некоторые n периодов эксплуатации ПО, его тестирования, отладки и тому подобное. Процесс роста надежности ПО описывается разностным уравнением

$$H_{n+1} - H_n = \sigma\beta(\alpha - H_n), \quad (9)$$

где σ – некоторая константа; α – ожидаемое число ошибок на бесконечно длинном интервале времени; β – интенсивность обнаружения ошибок.

Для численной оценки параметров α и β из (9) составим уравнение регрессии: $Y_n = A + BH_n$, где $Y_n = H_{n+1} - H_n$, $A = \delta\alpha\beta$, $B = -\delta\beta$.

Имея статистические данные по числу обнаруженных ошибок ($\hat{A}$ и $\hat{B}$), получаем оценки требуемых параметров из (9): $\hat{\alpha} = -\frac{\hat{A}}{\hat{B}}$ и $\hat{\beta} = -\frac{\hat{B}}{\delta}$.

Перейдем к процедурам второй группы, где разработана компонентная модель оценки надежности ПО, учитывающая несколько классов ошибок.

5. Формальное представление компонентной модели оценки надежности ПО.

Определение 2. Компонентную модель оценки надежности ПО, учитывающую структуру программных конструкций и классы ошибок, определим следующим образом. Имеются три класса ошибок:

1) $F_A = \{F_{A_1}, F_{A_2}, \dots, F_{A_k}\}$ – не критические, не воздействующие на выход своего и вход другого блока;

2) $F_B = \{F_{B_1}, F_{B_2}, \dots, F_{B_L}\}$ – переходящие, воздействующие на выход своего и вход другого блока;

3) $F_C = \{F_{C_1}, F_{C_2}, \dots, F_{C_M}\}$ – критические, воздействующие на выход своего и не воздействующие на вход другого блока.

Оценкой надежности компонента с различной программной конструкцией будем считать вероятность возникновения события, связанного с проявлением ошибки любого вышеуказанного класса, рассматривая программный компонент как систему «вход–структура–выход» $\Pr(I, O)$, $I \in F_B$, $O \in (F_B \cup F_C)$, причем $\sum_{O \in (F_B \cup F_C)} \Pr(I, O) = 1, \forall I \in F_B$.

6. Варианты расчетов возникновения события, сигнализирующего о появлении ошибки ПО.

В связи с заявленным типом модели «вход–структура–выход» рассмотрим варианты расчета вероятности возникновения события, вызванного ошибкой в зависимости от структуры программного компонента.

6.1. Последовательная структура.

На рисунке 5 показана последовательная структура исполнения алгоритмов $A_1, A_2, \dots, A_n$ программного компонента.

Вычислительная схема 1.

1. Генерация вероятностей $\Pr_{A_j}(I, F)$, $\Pr_{A_j}(I, O^*)$, $\Pr_{A_j}(I, C)$, $I \in F_B$, $F \in F_A$, $O^* \in F_B$, $C \in F_C$, $j = 1, \dots, n$.

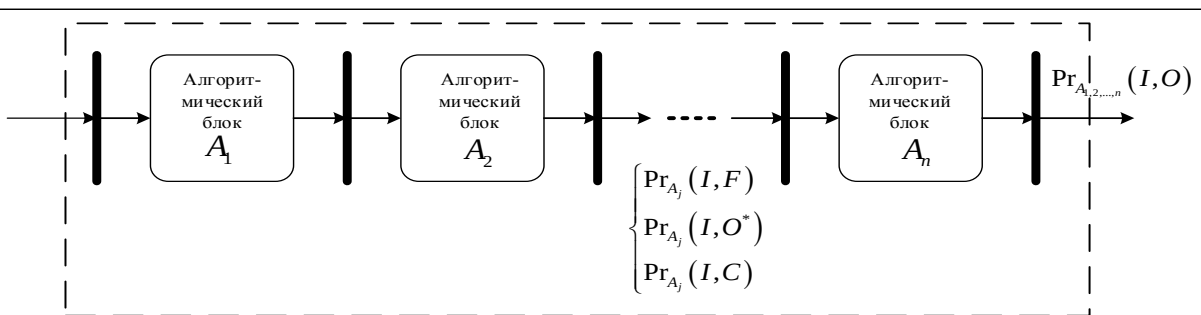


Рис. 5. Последовательная структура программного компонента

Fig. 5. A software component consecutive structure

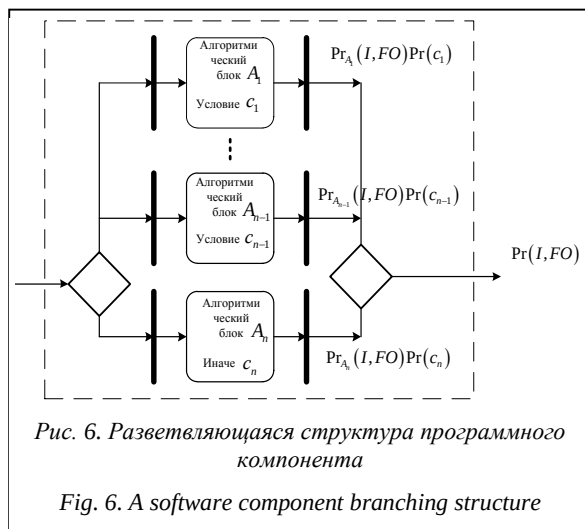
2. Для имитации возникновения некритической ошибки на выходе A_n : $\Pr_{A_{1,2,\dots,n}}(I, O) = \Pr_{A_j}(I, F)$, где $F \in F_A$.

3. Для имитации возникновения переходящей ошибки на выходе A_n : $\Pr_{A_{1,2,\dots,n}}(I, O) = \sum_{O^*} \Pr_{A_{1,2,\dots,n-1}}(I, O^*) \Pr_{A_n}(O^*, O)$, где $O^* \in F_B$.

4. Для имитации возникновения критической ошибки на выходе A_n : $\Pr_{A_{1,2,\dots,n}}(I, O) = \Pr_{A_{1,2,\dots,n-1}}(I, C) + \sum_{O^* \in F_B} \Pr_{A_{1,2,\dots,n-1}}(I, O^*) \Pr_{A_n}(O^*, O)$, где $O^* \in F_B$, $C \in F_C$.

6.2. Разветвляющаяся структура.

Если программный компонент имеет структуру разветвления с c_n ветвями, из которых первые $n-1$ являются ветвями «если-то», а n -я ветвь общим «иначе», то расчет будет производиться по вычислительной схеме 2 (рис. 6).



Вычислительная схема 2.

1. Генерация вероятностей $\Pr_{A_j}(I, FO)$, $I \in F_B$, $FO \in (F_B \cup F_C)$, $j = 1, \dots, n$, $\Pr(c_i)$, $i = 1, \dots, n-1$.

2. Для имитации возникновения ошибки рассчитывается $\Pr(I, O) = \sum_{i=1}^{n-1} \Pr_{A_i}(I, FO) \Pr(c_i) + \Pr_{A_n}(I, FO) \left(1 - \sum_{i=1}^{n-1} \Pr(c_i)\right)$.

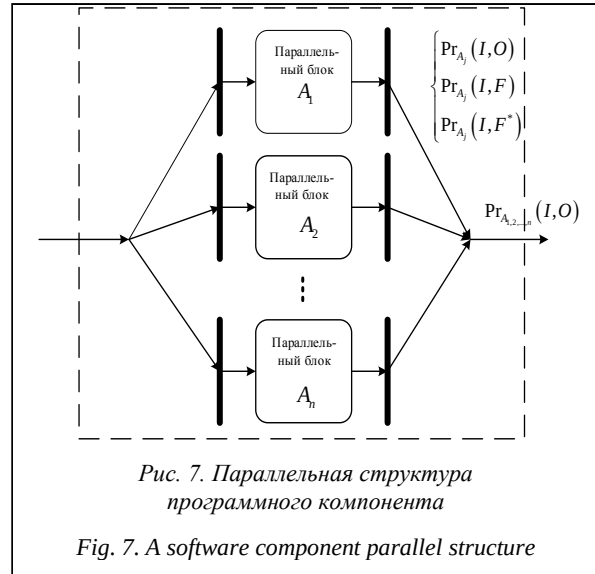
6.3. Циклическая структура.

Если программный компонент имеет циклическую структуру, ее можно трансформировать в n -кратное повторение последовательной структуры исполнения алгоритма $\underbrace{A_1, A_1, \dots, A_1}_{n \text{ раз}}$ програм-

ного компонента. Таким образом, для циклической структуры оказывается подходящей вычислительная схема из п. 6.1.

6.4. Параллельная структура.

Если программный компонент предусматривает параллельное исполнение n алгоритмов $A_1, A_2, \dots, A_n$, то моделируется возникновение событий,



связанных с классами некритичных и критичных ошибок. Структура исполнения алгоритмов в этом случае показана на рисунке 7.

Вычислительная схема 3.

1. Генерация вероятностей $\Pr_{A_j}(I, O)$, $\Pr_{A_j}(I, F)$, $\Pr_{A_j}(I, F^*)$, $I \in F_B$, $F^* \in F_A$, $F \in (F_A \cup F_C)$ $j = 1, \dots, n$.

2. Если $n-1$ параллельных ветвей имеют корректный выход, а n -я ветвь имеет некритическую ошибку, то рассчитывается $\Pr_{A_{1,2,\dots,n}}(I, O) = \Pr_{A_{1,2,\dots,n-1}}(I, O) \Pr_{A_n}(I, F^*)$.

3. Если $n-1$ параллельных ветвей имеют некритическую ошибку, а n -я ветвь имеет критическую ошибку либо наоборот, то рассчитывается $\Pr_{A_{1,2,\dots,n}}(I, O) = \left[\sum_{F \in (F_A \cup F_C)} \Pr_{A_{1,2,\dots,n-1}}(I, F) \right] \Pr_{A_n}(I, F)$.

На основании изложенного сделаем следующие выводы. В статье предложен дискретно-событийный подход к оценке надежности ПО, использующий покомпонентную технологию моделирования событий возникновения ошибок на входе и выходе программного компонента. Процессы обнаружения и устранения ошибок моделируются случайными точечными процессами, а времена обнаружения ошибок сопоставляются с событиями, при возникновении которых требуется рассчитать вероятностные оценки надежности программных компонентов, которые в зависимости от реализуемых ими алгоритмических структур имеют разные формулы для расчета.

Литература

1. Banks J. Discrete-event system simulation. Pearson Prentice Hall, 2005, 608 p.
2. Tyszer J. Object-oriented computer simulation of discrete-event systems. Springer US, 1999, 258 p.
3. Wainer G.A. Discrete event modeling and simulation: a practitioner's approach. CRC Press, 2009, 486 p.
4. Лоу А.М., Кельтон В.Д. Имитационное моделирование.

Классика CS. 3-е изд. СПб: Питер; Киев: BHV, 2004. 847 с.

5. Рыжиков Ю.И. Имитационное моделирование. Теория и технологии. М.: Альтекс-А, 2004. 384 с.

6. Финаев В.И. Алгоритмизация и имитационное моделирование с применением аппарата систем массового обслуживания: учеб. пособие. Таганрог: Изд-во ТРТУ, 2003. 155 с.

7. Таха Х.А. Введение в исследование операций. 7-е изд.; [пер. с англ.]. М.: Вильямс, 2005. 912 с.

8. Бражник А.Н. Имитационное моделирование: возможности GPSS WORLD. СПб: Реноме, 2006. 439 с.

9. Котов В.Е. Сети Петри. М.: Наука, 1984. 160 с.

10. Hruz B., Zhou M.C. Modeling and Control of Discrete-event Dynamic Systems: with Petri Nets and Other Tools. Springer London, 2007, 341 p.

11. Cassandras C.G., Lafortune S. Introduction to discrete event systems. 2nd Edition. Springer, 2008, 782 p.

12. Baccelli F., Cohen G., Olsder G.J., Quadrat J.-P. Synchronization and linearity: an algebra for discrete event systems. Chichester: Wiley, 1992, 514 p.

13. Gondran M., Minoux M. Graphs, Dioids and Semirings. New Models and Algorithms. Springer Science+Business Media, LLC, 2008, 384 p.

14. Heidergott B., Jan Olsder G., van der Woude J. Max Plus at work. Modeling and analysis of synchronized systems: A course on Max-Plus Algebra and Its Applications. Princeton Univ. Press, Princeton, 2006, 232 p.

15. Wainer G.A. Discrete Event Modeling and Simulation: A practitioner's approach. CRC Press, 2009, 486 p.

DOI: 10.15827/0236-235X.112.158-165

Received 09.09.15

SOFTWARE RELIABILITY EVALUATION BY DISCRETE-EVENT SIMULATION METHODS

(The work has been done with financial support from RFBR within the research projects no. 15-08-01886-a, 15-01-03067-a)

Butakova M.A., Dr.Sc. (Engineering), Professor, Dean, butakova@rgups.ru;

Guda A.N., Dr.Sc. (Engineering), Professor, Vice-rector, guda@rgups.ru;

Chernov A.V., Head of Chair, Dr.Sc. (Engineering), Professor, avche@yandex.ru;

Chubeyko S.V., Ph.D. (Engineering), greyc@mail.ru

(Rostov State Transport University,

PL Rostovskogo Strelkovogo Polka Narodnogo Opolcheniya 2, Rostov-on-Don, 344038, Russian Federation)

Abstract. The article considers discrete-event modeling and presents its distinctive features comparing to other types of modeling. The main difference is a lack of time reference, i.e. it is enough to keep the sequence of events, and thus the time interval between events isn't important. The authors give the definition of a discrete-event system model and add model time, which reproduce the course of events. The article solves the important problem of the event list generation in various ways: object-based and the process-based execution of events. Both ways are considered in detail with the illustration, algorithm and a program implementation element. Events can be united into groups called processes. The process-based modeling is more difficult than object-based as there is a process scheduler. The article also considers the assessment of software reliability based on discrete-event approach. This approach is based on the idea of software reliability growth. Bug scanning is modeled by a stochastic point process. A detected bug is eliminated, thereby the software becomes more reliable. Modeling has two parts: generation of the processes, which imitate introducing bugs in the software, and an assessment of component software system reliability. The paper considers options of calculating bug probability depending on program structure: consecutive, branching, cyclic and parallel. Each option has an illustration and a computation scheme. The cyclic scheme of a software component has the computation scheme of a consecutive component as it is some kind of same-type repetitions of a software component consecutive structure.

Keywords: discrete-event simulation, event-based modeling, process-based modeling, software reliability, software structure.

References

1. Banks J. *Discrete-event System Simulation*. Pearson Prentice Hall Publ., 2005, 608 p.
2. Tyszer J. *Object-oriented Computer Simulation of Discrete-event Systems*. Springer US Publ., 1999, 258 p.
3. Wainer G.A. *Discrete Event Modeling and Simulation: A Practitioner's Approach*. CRC Press, 2009, 486 p.
4. Lou A., Kelton V. *Simulation. Classics CS*. 3th ed., St.-Petersburg, Piter, Kiev, BHV Publ., 2004, 847 p.
5. Ryzhikov U. *Imitatsionnoe modelirovanie. Teoriya i tekhnologii* [Simulation. Theory and Technology]. Moscow, Altex-A Publ., 2004, 384 p.
6. Finaev V.I. *Algoritmizatsiya i imitatsionnoe modelirovanie s primeneniem apparata sistem massovogo obsluzhivaniya* [Algorithmic and Simulation using Queueing Theory]. Study guide, Taganrog, Taganrog State Univ. of Radio Engineering Publ., 2003, 155 p.
7. Taha H.A. *Operations Research: an Introduction*. 7th ed. 2002, Prentice Hall Publ., 848 p. (Russ. ed.: Vilyams Publ., 2005, 912 p.).
8. Brazhnik A. *Imitatsionnoe modelirovanie: vozmozhnosti GPSS WORLD* [Simulation: GPSS WORLD capabilities]. St.-Petersburg, Renome Publ., 2006, 439 p.
9. Kotov V. *Seti Petri* [Petri Nets]. Moscow, Nauka Publ., 1984, 160 p.
10. Hruz B., Zhou M.C. *Modeling and Control of Discrete-event Dynamic Systems: with Petri Nets and Other Tools*. Springer London Publ., 2007, 341 p.
11. Cassandras C.G., Lafortune S. *Introduction to Discrete Event Systems*. 2nd Edition. Springer Publ., 2008, 782 p.
12. Baccelli F., Cohen G., Olsder G.J., Quadrat J.-P. *Synchronization and linearity: An algebra for discrete event systems*. Chichester, Wiley Publ., 1992, 514 p.
13. Gondran M., Minoux M. *Graphs, Dioids and Semirings. New Models and Algorithms*. Springer Science+Business Media Publ., 2008, 384 p.
14. Heidergott B., Jan Olsder G., van der Woude J. *Max Plus at work. Modeling and analysis of synchronized systems: A course on Max-Plus Algebra and Its Applications*. Princeton Univ. Press, Princeton, 2006, 232 p.
15. Wainer G.A. *Discrete Event Modeling and Simulation: A Practitioner's Approach*. CRC Press, 2009, 486 p.

УДК 004.056

Дата подачи статьи: 28.09.15

DOI: 10.15827/0236-235X.112.166-174

МЕТОДИЧЕСКИЙ АППАРАТ АНАЛИЗА И СИНТЕЗА КОМПЛЕКСА МЕР РАЗРАБОТКИ БЕЗОПАСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.В. Барабанов, к.т.н., директор, ab@snpro.ru

(НПО «Эшелон», ул. Электроставская, 24, г. Москва, 107023, Россия);

А.С. Марков, д.т.н., профессор, старший научный сотрудник, a.markov@bmstu.ru;

В.А. Цирлов, к.т.н., доцент, v.tsirlov@bmstu.ru

*(Московский государственный технический университет им. Н.Э. Баумана,
ул. 2-я Бауманская, 5, г. Москва, 105005, Россия)*

Рассмотрены актуальные вопросы стандартизации серийного производства безопасных программных изделий. Исследованы организационно-технические меры по снижению количества уязвимостей при разработке и сопровождении ПО функционирования автоматизированных систем в защищенном исполнении. Проведена систематизация стандартов и рекомендаций в области разработки безопасного ПО. Выполнен анализ применимости существующих методических подходов к разработке безопасного ПО при проведении оценки соответствия требованиям безопасности информации, в том числе при сертификации программных средств. Показана целесообразность гармонизации разрабатываемых нормативных требований и практических мер с методологиями международных стандартов по линии ISO 15408 и ISO 12207. Введено понятие безопасного ПО. Разработан базовый набор требований, позволяющий проводить и оценку соответствия процессов разработки ПО требованиям к безопасному ПО. При этом обосновано, что набор требований должен опираться прежде всего на принятые политики безопасности и актуальные угрозы. Приведен пример разрабатываемых требований. Разработана оригинальная концептуальная модель анализа и синтеза комплекса мер разработки безопасного ПО, опирающаяся на набор формируемых требований. Показано, что концептуальная модель предоставляет разработчикам ПО возможность научно обоснованного выбора мер разработки ПО. Разработана общая методика выбора комплекса мер безопасной разработки ПО. Представлены косвенные подтверждения эффективности предлагаемого подхода. Отмечено, что предложенный подход лег в основу разработки национального стандарта в области разработки и производства безопасного ПО.

Ключевые слова: защита информации, безопасность программ, безопасный жизненный цикл, безопасное производство программ, безопасная программная инженерия, уязвимости, общие критерии, механизмы безопасности приложений.

Современное состояние *информационной безопасности* (ИБ) компьютерных систем характеризуется превалированием угроз, связанных с наличием как незакрытых известных, так и неизвестных уязвимостей. В то же время в стране фактически отсутствует регламентация безопасной разработки программных средств, ориентированная на снижение количества уязвимостей ИБ, формируемых на различных этапах жизненного цикла программ. В этом плане остается нерешенным вопрос проведения как спецэкспертиз (лицензирования) предприятий, так и проверки производств (оценки соответствия) серийных программных изделий по требованиям безопасности информации.

Это определяет исследовательскую задачу, состоящую в анализе и синтезе комплекса мер по разработке безопасного ПО, решение которой представлено в данной статье.

Содержательная постановка задачи исследования

Мерами по безопасной разработке ПО будем считать организационно-технические меры, направленные на создание программ с минимально возможным количеством уязвимостей и формирование среды обеспечения оперативного устранения уязвимостей, выявленных пользователями. Опыт

крупных разработчиков ПО показал, что внедрение подобных мер в жизненный цикл его разработки позволяет сократить число уязвимостей ПО в среднем на 80 % [1].

Безопасным будем считать ПО, разработанное с использованием совокупности указанных мер.

Цель исследования состоит в создании концептуального аппарата, позволяющего разработчикам ПО обоснованно формировать множество мер разработки безопасного ПО и с привлечением независимых организаций проводить оценку соответствия применяемых мер требованиям по разработке безопасного ПО.

В соответствии с целью необходимо поставить следующие частные задачи исследования:

- анализ существующих мер, направленных на уменьшение количества уязвимостей в разрабатываемом ПО, и их применимости при проведении оценки соответствия ПО;
- формирование базового набора требований к разработке безопасного ПО, позволяющего проводить оценку соответствия (в частном случае сертификацию) ПО данным требованиям;
- разработка концептуальной модели анализа и синтеза комплекса мер разработки безопасного ПО.

В качестве ограничения области исследования следует определить необходимость гармонизации

получаемых решений с современной нормативной базой оценки соответствия в области ИБ (линейка ISO 15408/18045, Common Criteria) [2] и жизненного цикла программ (ISO 12207) [3].

Анализ стандартов по безопасной разработке

В настоящее время известен ряд корпоративных, отраслевых и международных стандартов, содержащих описание мер и механизмов, направленных на разработку безопасного ПО, например, ISO 15408 [4–6], ISO 27034-1, ISO TR 24772 [7], Microsoft Security Development Life Cycle [1], Cisco Security Development Life Cycle, OpenSAMM [8], OWASP CLASP [9].

Результаты сравнительного анализа указанных стандартов представлены в таблице 1.

На основании проведенного анализа сделаны следующие выводы.

1. Существуют достаточно детально проработанные стандарты на разработку безопасного ПО, направленные, как правило, на следующие цели:

- создание ПО с минимально возможным количеством уязвимостей и формирование среды для обеспечения оперативного устранения выявленных пользователем ПО проблем (уязвимостей ПО);
- реализация ПО элементов политики обеспечения безопасности среды (места) применения ПО

(политики ИБ организации или страны, где применяется ПО).

2. Для создания ПО с минимальным количеством уязвимостей и формирования среды устранения выявленных проблем рассмотренные документы предлагают использовать на различных стадиях жизненного цикла ПО совокупность мер разработки безопасного ПО. Предлагаемая номенклатура мер разработки безопасного ПО, как правило, является стандартной и содержит меры, связанные, например, с моделированием угроз, проведением статического и динамического анализа, тестирования на проникновение [10, 11].

3. Рассмотренные документы не содержат четко определенного аппарата, который можно было бы использовать для оценки соответствия процессов разработки ПО требованиям разработки безопасного ПО, например в рамках его сертификации. Следует отметить, что международный стандарт ISO 15408 в настоящее время широко используется при проведении сертификации ПО по требованиям безопасности информации, но использование положений только этого документа при разработке и последующей оценке соответствия безопасного ПО не является достаточным, поскольку, во-первых, стандарт применяется только для ПО с функциями безопасности, иначе говоря, для ПО, в котором функции безопасности не реализованы, не может быть написано требуемое стандартом зада-

Таблица 1

Наличие в стандартах мер и механизмов, направленных на безопасность

Table 1

Security measures and mechanisms in standards

Мера и механизм	Стандарт						
	ISO 15408	Microsoft SDL	Cisco SDL	Open SAMM	OWASP CLASP	ISO TR 24772	ISO 27034-1
Обучение сотрудников организации-разработчика разработке безопасного ПО	-	+	+	+	-	-	-
Обеспечение безопасности инфраструктуры разработки ПО	+	-	-	-	-	-	-
Управление конфигурацией разрабатываемого ПО	+	-	-	-	-	-	-
Моделирование угроз безопасности информации, источником которых является ПО	+	+	+	+	+	-	-
Определение требования по разработке безопасного ПО	+	+	+	+	+	-	-
Использование стандарта на оформление исходного кода программы (правил и рекомендаций, направленных на минимизацию количества потенциально уязвимых конструкций в исходном коде программы)	-	+	+	+	+	+	-
Проведение статического анализа исходного кода программы	-	+	+	+	+	-	-
Проведение динамического анализа динамического кода программы	-	+	+	+	+	-	-
Проведение экспертизы исходного кода программ в ручном режиме	-	+	+	+	+	-	-
Проведение анализа уязвимостей	-	+	+	+	+	-	-
Обеспечение безопасности поставки ПО пользователю	+	+	+	+	-	-	-
Устранение уязвимостей ПО, выявляемых в процессе функционирования ПО	+	+	+	+	+	-	-
Возможность использования документа при оценке соответствия ПО требованиям, предъявляемым к разработке безопасного ПО	+	-	-	-	-	-	-
Наличие методики выбора подмножества мер разработки безопасного ПО	+	+	+	+	-	-	+
Согласованность с процессами жизненного цикла ПО, определенными ISO 12207	-	-	-	-	-	-	+

ние по безопасности и, соответственно, проведена оценка соответствия; во-вторых, предлагаемая в третьей части стандарта номенклатура мер разработки ПО не обладает полнотой: в частности, отсутствуют меры, связанные с проведением статического и динамического анализа, обучением сотрудников и др.

Базовый набор требований

Как известно, современное развитие отечественной системы сертификации средств защиты информации по требованиям безопасности информации связано с апробацией методологии Common Criteria (ISO 15408) [12]. По этой причине в данном исследовании было принято решение об использовании аппарата ISO 15408 при разработке концептуальной модели анализа и синтеза мер разработки безопасного ПО.

На рисунке 1 представлена последовательность формирования множеств функций безопасности ПО (объекта оценки) и мер обеспечения доверия к нему в соответствии с методологией ISO 15408.

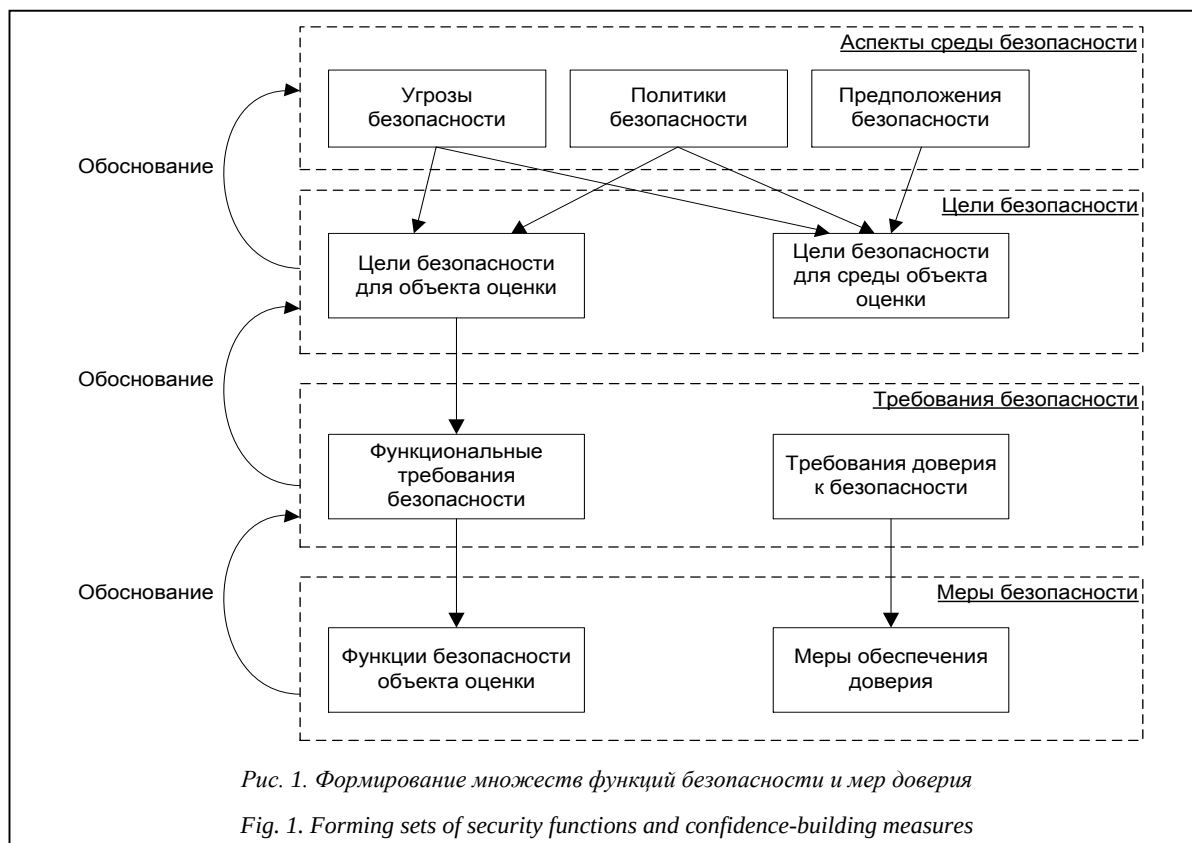
При определении среды безопасности ПО приводится описание следующих аспектов безопасности среды, в которой предполагается использовать ПО:

- предположений безопасности, содержащих аспекты безопасности среды, в которой будет использоваться ПО или предполагается к использованию;

- угроз безопасности информации, касающихся активов, против которых требуется защита средствами ПО или его среды;

- политик безопасности организации, идентифицирующих и при необходимости объясняющих положения политики безопасности организации или правила, которым должно подчиняться ПО.

Цели безопасности отражают изложенное намерение противостоять всем установленным угрозам ИБ, а также охватывать все предположения безопасности и установленную политику безопасности организации. При изложении требований безопасности ПО должны быть определены функциональные требования (например, требования к идентификации/аутентификации или разграничению доступа) и требования доверия, которым должны удовлетворять ПО и процесс его разработки. Выбор функциональных требований безопасности и требований доверия к безопасности осуществляется из каталогов требований, представленных во второй и третьей частях стандарта. С целью удовлетворения идентифицированных функциональных требований безопасности и требований доверия формулируются перечни функций безопасности ПО и мер разработки ПО. Все идентифицированные множества (аспекты среды безопасности, цели, требования, меры) должны быть согласованы друг с другом (данная информация и необходимые пояснения приводятся, как правило, в задании по безопасности).



Для адаптации представленного на рисунке 1 порядка формирования и обоснования перечней функций безопасности ПО и мер разработки ПО, а также достижения цели настоящего исследования был получен базовый набор требований по разработке безопасного ПО. При формировании базового набора требований по разработке безопасного ПО учитывались следующие особенности.

1. С целью обеспечения практического применения и эффективного выполнения требований разрабатываемый набор должен согласовываться с международным стандартом ISO 12207, который описывает процессы жизненного цикла ПО [3, 13].

2. Поскольку известно, что стоимость устранения уязвимостей ПО в более поздних процессах жизненного цикла ПО (например, в процессе тестирования или поддержки ПО) выше, разрабатываемый набор требований должен обеспечить внедрение необходимых мер на самых ранних стадиях разработки ПО [14, 15].

3. Для решения задачи разработки аппарата проведения оценки соответствия ПО разрабатываемым требованиям при формировании отдельных требований должны быть учтены аспекты, связанные с формированием требований к документальным свидетельствам выполнения требования и к действиям оценщика (например, эксперта испытательной лаборатории), выполняемым в ходе оценки соответствия ПО требованиям к безопасному ПО [7].

При выполнении исследования с целью учета выявленных особенностей было предложено использовать следующий набор параметров при описании требования по разработке безопасного ПО:

- название требования;
- уникальный идентификатор требования;
- ссылка на процесс жизненного цикла ПО, установленный стандартом ISO 12207, в ходе которого может быть выполнено требование;
- цель в области разработки безопасного ПО, достигаемая разработчиком при выполнении данного требования;
- элементы действий разработчика, содержащие описание действий разработчика ПО, направленных на удовлетворение требования;
- элементы содержания и представления документированных свидетельств выполнения требования;
- элементы действий оценщика, содержащие описание действий оценщика, направленных на независимую проверку выполнения разработчиком ПО требования;
- примечание с пояснительным текстом.

В ходе выполнения исследования на основе существующих стандартов и методологий разработки безопасного ПО синтезировано множество $B = \{\beta_1, \beta_2, \dots, \beta_{24}\}$ требований по разработке безопасного ПО (см. табл. 2).

Так как статический анализ безопасности программ является наиболее результативным (к сожа-

лению, весьма трудоемким [16–18]), целесообразно следовать следующим разработанным требованиям КК-4 (β_7).

Название: статический анализ исходного кода программы.

Идентификатор требования: КК-4.

Процесс жизненного цикла ПО: процессы конструирования и комплексирования ПО.

Цель: выявление и устранение потенциально уязвимых конструкций в исходном коде программы, а также формирование исходных данных для выполнения задач динамического анализа и тестирования на проникновение в рамках процесса квалификационного тестирования ПО.

Элементы действий разработчика: разработчик ПО должен проводить статический анализ исходного кода программы с целью выявления потенциально уязвимых конструкций в исходном коде программы; статический анализ исходного кода программы следует проводить в отношении заимствованных у сторонних разработчиков ПО компонентов, если для них доступен исходный код программы; по результатам статического анализа исходного кода программы может проводиться доработка программы; при отсутствии необходимости в такой доработке или невозможности доработки программы разработчик должен документировать обоснование этого факта.

Элементы содержания и представления документированных свидетельств – документированные свидетельства статического анализа исходного кода программы должны содержать

- сведения о периодичности проведения статического анализа исходного кода программы;
- наименование и идентификационные признаки инструментальных средств, используемых для проведения статического анализа исходного кода программы;
- список выявленных потенциально уязвимых конструкций в исходном коде программы (при выявлении), описание действий, направленных на их устранение, или обоснование невозможности или отсутствия необходимости в доработке программы.

Элементы действий оценщика: оценщик должен исследовать представленные свидетельства и подтвердить, что они удовлетворяют предъявляемым требованиям, а также сделать независимое заключение, что разработчик выполняет статический анализ исходного кода программы по результатам опроса работников организации-разработчика ПО, имеющих отношение к разработке ПО, и анализ среды разработки ПО.

Примечание: статический анализ исходного кода программы выполняется разработчиком ПО или сторонними организациями, обладающими компетенцией в области выявления уязвимостей ПО, для актуальной версии кода программы; стати-

Таблица 2

Предлагаемый базовый набор требований по разработке безопасных программ

Table 2

The suggested core set of requirements to develop secure programs

Идентификатор требования	Краткое содержание требования	Процесс жизненного цикла ПО (ISO 12207)
АТ-1 (β_1)	Разработчик ПО должен определить требования в части разработки безопасного ПО, предъявляемые к разрабатываемому ПО	Процесс анализа требований к ПО
ПА-1 (β_2)	Разработчик ПО должен выполнить моделирование угроз безопасности информации, источником которых является ПО, с целью выявления потенциальных угроз безопасности информации и обоснования требований к ПО	Процесс проектирования и детального проектирования архитектуры ПО
ПА-2 (β_3)	Архитектура ПО (логическая структура ПО, в которой идентифицированы компоненты, их интерфейсы и концепция взаимодействия между ними) должна уточняться с учетом необходимости нейтрализации потенциальных угроз безопасности информации, которые были выявлены на этапе моделирования угроз безопасности информации, и выполнение требований в части разработки безопасного ПО, установленных в процессе анализа требований к ПО	Процессы проектирования и детального проектирования архитектуры ПО
КК-1 (β_4)	Разработчик ПО должен идентифицировать каждое инструментальное средство, используемое при разработке ПО	Процессы конструирования и комплексирования ПО
КК-2 (β_5)	Разработчик ПО должен создавать программу на основе архитектуры ПО, определенной в ходе выполнения процессов проектирования и детального проектирования архитектуры ПО	Процессы конструирования и комплексирования ПО
...		
МДК-3 (β_{18})	Система управления конфигурацией ПО должна идентифицировать элементы конфигурации, которые связаны с реализацией функций безопасности ПО	Процесс менеджмента документации и конфигурации ПО
МДК-4 (β_{19})	Система управления конфигурацией ПО должна предоставлять средства для определения всех элементов конфигурации, на которые воздействует модификация данного элемента конфигурации	Процесс менеджмента документации и конфигурации ПО
МИ-1 (β_{20})	Разработчиком ПО должны применяться технические и организационные меры, обеспечивающие защиту от несанкционированного доступа к элементам конфигурации	Процесс менеджмента инфраструктуры
МИ-2 (β_{21})	Разработчиком ПО должны применяться технические и организационные меры, обеспечивающие резервное копирование и восстановление элементов конфигурации	Процесс менеджмента инфраструктуры
МИ-3 (β_{22})	Разработчиком ПО должны применяться технические и организационные меры, обеспечивающие регистрацию всех событий, связанных с фактами изменения элементов конфигурации, в журнале регистрации событий	Процесс менеджмента инфраструктуры
МЛР-1 (β_{23})	Разработчик ПО должен проводить периодическое обучение сотрудников с целью повышения их осведомленности в области разработки безопасного ПО	Процесс менеджмента людских ресурсов
МЛР-2 (β_{24})	Разработчик ПО должен проводить периодический анализ программы обучения сотрудников для установления ее пригодности, адекватности и результативности для достижения установленных целей в области разработки безопасного ПО	Процесс менеджмента людских ресурсов

ческий анализ исходного кода программы позволяет выполнить поиск потенциально уязвимых конструкций в исходном коде программы, которые могут привести к наличию уязвимости ПО, а также проверять соответствие исходного кода программы принятому в организации стандарту оформления исходного кода программы.

Представленный базовый набор требований разработки безопасного ПО использовался при создании концептуальной модели анализа и синтеза комплекса мер разработки безопасного ПО.

Концептуальная модель

Введем следующие множества для описания концептуальной модели анализа и синтеза мер разработки безопасного ПО.

При описании среды разработки ПО следует определить:

– множество угроз ИБ, которые являются актуальными для среды разработки ПО: $U = \{u_1, u_2, \dots, u_a\}$;

– множество идентифицированных положений политик ИБ, которым должна соответствовать среда разработки ПО: $P = \{p_1, p_2, \dots, p_b\}$.

Заметим, что в соответствии с ISO 15408 угрозы безопасности информации можно описать в неформальном (содержательном) стиле с использованием таких характеристик, как аннотация угрозы ИБ; источник угрозы ИБ; способ реализации угрозы ИБ; используемые уязвимости; вид информационных ресурсов среды разработки ПО, потенциально подверженных угрозе ИБ; нарушаемое свойство безопасности информационных ресурсов

среды разработки ПО; возможные последствия реализации угрозы ИБ.

При описании среды безопасности согласно ISO 15408 (см. рис. 1) предусмотрено использование дополнительного множества предположений безопасности. Однако, так как при проведении анализа и синтеза комплекса мер разработки безопасного ПО среда разработки ПО точно идентифицирована и определена, формирование перечня предположений является избыточным.

Для достижения идентифицированных политик безопасности и нейтрализации идентифицированных угроз ИБ следует сформировать множество целей разработки безопасного ПО: $O = \{o_1, o_2, \dots, o_c\}$.

Разработчик из определенного ранее множества B для своей конкретной инфраструктуры может выбрать подмножество требований, подлежащих выполнению:

$$R = \{r_1, r_2, \dots, r_d\}, R \subseteq B, d \leq 24.$$

Для выполнения идентифицированных требований разработчик ПО может сформировать и реализовать в среде разработки ПО комплекс мер разработки безопасного ПО, а именно множество $C = \{c_1, c_2, \dots, c_e\}$.

Введем следующие отображения:

$F_0: U \cup P \rightarrow O$ – процедура формирования целей разработки безопасного ПО;

$F_R: B \cup O \rightarrow R$ – процедура выбора требований по разработке безопасного ПО;

$F_C: R \rightarrow C$ – процедура синтеза мер разработки безопасного ПО.

Разработанная концептуальная модель характеризуется кортежем $\langle B, U, P, O, R, F_0, F_R, F_C \rangle$.

При разработке модели предполагаем, что все требования являются нефункциональными, иначе говоря, по сути это аналоги требований доверия по ISO 15408-3 (рис. 2).

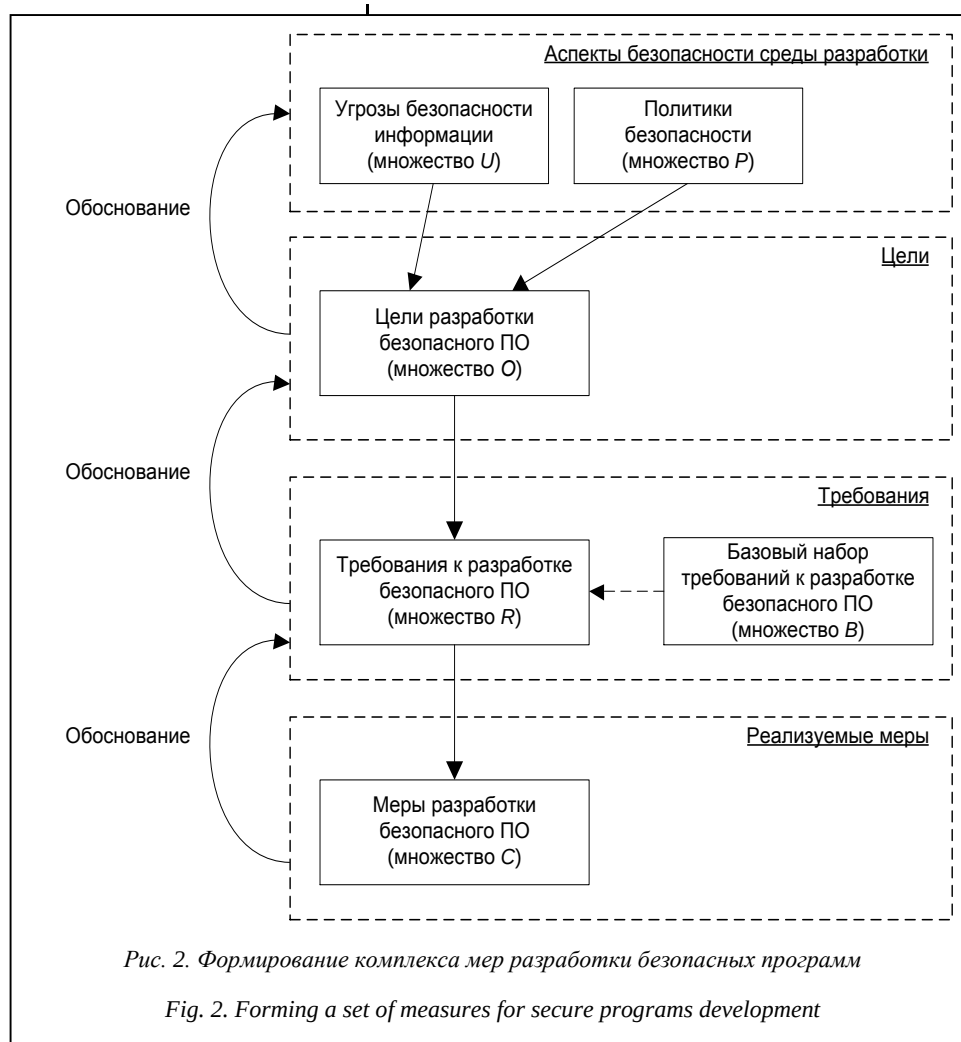


Рис. 2. Формирование комплекса мер разработки безопасных программ

Fig. 2. Forming a set of measures for secure programs development

В ходе проведения исследования была разработана общая методика анализа и синтеза мер разработки безопасного ПО, включающая выполнение следующих этапов и шагов.

Этап 1. Идентификация и описание аспектов безопасности среды разработки ПО, включающих следующие шаги.

Шаг 1. Формирование множества $U = \{u_1, u_2, \dots, u_a\}$ угроз ИБ, которые являются актуальными для среды разработки ПО.

Шаг 2. Формирование множества $P = \{p_1, p_2, \dots, p_b\}$ положений политик ИБ, которым должна соответствовать среда разработки ПО. В качестве источников для формирования множества политик безопасности могут использоваться требования законов, нормативных правовых актов, отраслевых стандартов, перечень требований пользователя, сценарии использования ПО.

Этап 2. Формирование и обоснование множества целей разработки безопасного ПО, включающих следующие шаги.

Шаг 3. Формирование множества целей разработки безопасного ПО: $o_i = F_o(u_j, p_l); i = \overline{1; c}$.

Шаг 4. Обоснование полноты и достаточности сформулированного множества целей разработки безопасного ПО. Обоснование должно показывать, что изложенные цели охватывают все идентифицированные аспекты безопасности (множества угроз ИБ и положений политик ИБ) среды разработки ПО. Данная демонстрация выполняется следующим образом:

а) обоснование полноты с использованием перекрестных ссылок – цели разработки безопасного ПО, направленные на учет идентифицированных множеств угроз ИБ и положений политики ИБ, целесообразно оформить в виде таблицы, которая должна наглядно показывать, что

- каждая цель охватывает, по крайней мере, одну угрозу ИБ или положение политики ИБ;
- каждая угроза ИБ или положение политики по безопасности охвачены, по крайней мере, одной целью разработки безопасного ПО;

б) обоснование достаточности множества целей разработки безопасного ПО – необходимо продемонстрировать, что множества целей разработки безопасного ПО достаточно для учета всех аспектов среды разработки ПО; для этого таблицу соответствия целей и аспектов среды разработки ПО необходимо дополнить следующим неформальным пояснением:

- для каждой угрозы ИБ необходимо продемонстрировать, что изложенные цели позволяют выполнить нейтрализацию данной угрозы ИБ;
- для каждого положения политики ИБ необходимо показать, каким образом изложенные цели обеспечивают ее выполнение.

Этап 3. Выбор и обоснование требований по разработке безопасного ПО, включающие следующие шаги.

Шаг 5. Формирование множества требований по разработке безопасного ПО на основе базового набора требований с учетом необходимости достижения идентифицированных целей: $r_m = F_R(b_n, o_i)$; $m = \overline{1, c}$.

Шаг 6. Обоснование полноты и достаточности выбранного множества требований по разработке безопасного ПО. Обоснование выполняется аналогично описанному в шаге 4, с использованием перекрестных ссылок и неформального пояснительного теста.

Этап 4. Синтез и обоснование мер разработки безопасного ПО.

Шаг 7. На основе множества идентифицированных требований по разработке безопасного ПО и информации о конкретной среде разработки ПО, технологиях и инструментальных средствах, применяемых при разработке, необходимо сформулировать множество мер разработки безопасного ПО, которые планируется применять в среде разработки ПО: $c_q = F_C(r_m)$; $q = \overline{1; e}$.

Шаг 8. Обоснование полноты и достаточности синтезированного множества мер разработки без-

опасного ПО. Обоснование выполняется аналогично описанному в шаге 4, с использованием перекрестных ссылок и неформального пояснительного теста.

Переход к математическим моделям

Предложенная концептуальная модель также может быть взята за основу при разработке математических моделей безопасной разработки, например, многофакторных моделей [19]:

$$P = \sum_{i=1}^K k_i p_i + \sum_{i \neq j}^K k_{ij} p_i p_j + \sum_{i=1}^K k_{ii} p_i^2 + \dots,$$

где k_i – коэффициент i -й меры; $K \leq 24$ – число мер.

Следует отметить, что в работе не ставилась цель получить значения указанных математических показателей, однако эффективность предложенного концептуального подхода была подтверждена результатами сравнения уровня уязвимости программных продуктов, разработанных в организациях, имеющих авторитетные зарубежные сертификаты на системы менеджмента ИБ, и продуктов, разработанных иными разработчиками ПО. Анализ статистики за 2008–2015 гг. показал, что обобщенный показатель числа уязвимостей в продуктах первой категории ниже в 5 раз, чем аналогичный показатель у продуктов второй категории [20].

В заключение отметим, что в ходе проведенных исследований были получены следующие основные результаты.

Выполнен анализ существующих мер и подходов, направленных на уменьшение количества уязвимостей в разрабатываемом ПО, и их применимости при проведении оценки соответствия ПО требованиям по безопасности информации. В результате установлено, что на сегодняшний момент существует ряд стандартов различного уровня, посвященных разработке безопасного ПО. Вместе с тем нет единого подхода, оформленного в форме нормативного документа, обеспечивающего

- интеграцию с существующими стандартами, определяющими процессы жизненного цикла программ, например, определенными ISO 12207;
- обоснованный выбор совокупности мер разработки безопасного ПО;
- оценку соответствия процессов создания программных изделий требованиям разработки безопасного ПО, в первую очередь, в рамках обязательной сертификации серийных программных средств защиты информации (при аттестации производства) и обязательного лицензирования таких видов деятельности, как производство программных средств защиты информации.

На основе анализа, систематизации и обобщения стандартов в области разработки безопасного ПО разработан базовый набор из 24 требований по разработке безопасного ПО. Отличительные особенности разработанного базового набора требований:

– наличие перекрестных ссылок на процессы жизненного цикла ПО, определенные ISO 12207, что обеспечивает более простую интеграцию в уже существующие в организации-разработчике ПО процессы разработки ПО;

– наличие требований к документированным свидетельствам выполнения требований и к действиям независимого оценщика, что позволяет детерминировать процесс независимой оценки соответствия процессов разработки ПО требованиям разработки безопасного ПО, например, в рамках сертификации ПО.

Предложена концептуальная модель анализа и синтеза мер разработки безопасного ПО, которая обеспечивает возможность обоснованного выбора мер разработки безопасного ПО и обладает свойством согласованности со стандартами линейки Common Criteria. Это позволяет выполнить внедрение предложенной в рамках модели методики анализа и синтеза комплекса мер разработки безопасного ПО в организациях, уже имеющих опыт проведения сертификации разрабатываемого ПО в соответствии с методологией ISO 15408.

Концептуальная модель анализа и синтеза комплекса мер разработки безопасного ПО использовалась при разработке проекта национального стандарта ГОСТ Р «Защита информации. Разработка безопасного программного обеспечения. Общие требования», прошедшего экспертизу в рамках работы Технического комитета по стандартизации ТК-362 «Защита информации».

Литература

- Howard M., Lipner S. The security development lifecycle: a process for developing demonstrably more secure software. Microsoft Press, 2006, 352 p.
- Markov A., Luchin D., Rautkin Y., Tsirlov V. Evolution of a radio telecommunication hardware-software certification paradigm in accordance with information security requirements. Proc. 11th Intern. Siberian Conf. on Control and Communications (Omsk, Russia, May 21–23, 2015). SIBCON-2015. IEEE, 2015, pp. 1–4; URL: <http://dx.doi.org/10.1109/SIBCON.2015.7147139> (дата обращения: 20.09.2015).
- Липаев В.В. Развитие базовых стандартов программной инженерии // Программная инженерия. 2012. № 6. С. 2–7.
- Biró M., Molnár B. Synergies between the common criteria and process improvement. LNCS, 2007, no. 4764, pp. 31–45; URL: http://dx.doi.org/10.1007/978-3-540-75381-0_4 (дата обращения: 20.09.2015).
- Kara M. Review on Common Criteria as a secure software development model. IJCSIT, 2012, vol. 4, no. 2 (Apr. 2012), pp. 83–94. DOI = <http://dx.doi.org/10.5121/ijcsit.2012.4207>.
- Lee M.-G., Sohn H.-J., Kim B.-M., Kim J. B. A Study on secure SDL Specialized in Common Criteria. ASTL, 2015, no. 93, pp. 11–23.
- Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhlov I. Synthesis of Secure Software Development controls. Proc. 8th Intern. Conf. SIN '15 (Sochi, Russian Federation, September 08–10, 2015). ACM, NY, USA, 2015, pp. 93–97; URL: <http://dx.doi.org/10.1145/2799979.2799998> (дата обращения: 20.09.2015).
- Барabanov A.B. Стандартизация процесса разработки безопасных программных средств // Вопросы кибербезопасности. 2013. № 1 (1). С. 37–41.
- De Win B., Scandariato R., Buyens K., Grégoire J., Joosen W. On the Secure Software Development Process: CLASP, SDL and touchpoints compared. information and software technology, 2009, vol. 51, no. 7 (Jul. 2009), pp. 1152–1171; URL: <http://dx.doi.org/10.1016/j.infsof.2008.01.010> (дата обращения: 20.09.2015).
- Essafi M., Ghezala H.B. Meta-modeling based secure software development processes. IJSSE, 2014, no. 5 (3) (Jul.–Sep. 2014), pp. 56–74; URL: <http://dx.doi.org/10.4018/ijssse.2014070104> (дата обращения: 20.09.2015).
- Fal' A.M. Standardization in information security management. Cybernetics and Systems Analysis, 2010, vol. 46, no. 3 (May 2010), pp. 512–515; URL: <http://dx.doi.org/10.1007/s10559-010-9227-9> (дата обращения: 20.09.2015).
- Barabanov A., Markov A. Modern trends in the regulatory framework of the information security compliance assessment in Russia based on Common Criteria. Proc. 8th Intern. Conf. SIN '15 (Sochi, Russian Federation, September 08–10, 2015). ACM, NY, USA, 2015, pp. 30–33; URL: <http://dx.doi.org/10.1145/2799979.2799980> (дата обращения: 20.09.2015).
- Невлюдов И.Ш., Андрусевич А.А., Евсеев В.В. Анализ жизненного цикла разработки программного обеспечения для корпоративных информационных систем // Восточ.-Европ. журн. передовых технологий. 2010. Т. 6. № 8 (48). С. 25–27.
- Рибер Г., Малмквист К., Щербаков А. Многоуровневый подход к оценке безопасности программных средств // Вопросы кибербезопасности. 2014. № 1 (2). С. 36–39.
- Viega J., McGraw G. Building Secure Software: how to avoid security problems the Right Way. Addison-Wesley Professional, 2001, 528 p.
- Аветисян А.И., Белеванцев А.А., Чукляев И.И. Технологии статического и динамического анализа уязвимостей программного обеспечения // Вопросы кибербезопасности. 2014. № 3 (4). С. 20–28.
- Демин А.А., Карпунин А.А., Ганев Ю.М. Методы верификации и валидации сложных программных систем // Программные продукты и системы. 2014. № 108. С. 229–233.
- Ardi S., Shahmehri N. Introducing vulnerability awareness to Common Criteria's security targets. Proc. 4th Intern. Conf. on Software Engineering Advances (20–25 Sept. 2009, Porto). ICSEA '09. IEEE Xplore Digital Library, 2009, pp. 419–424; URL: <http://dx.doi.org/10.1109/ICSEA.2009.67> (дата обращения: 20.09.2015).
- Марков А.С., Ларионцева Е.А., Стельмашук Н.Н. Многофакторные модели планирования сертификационных испытаний программных средств защиты информации // Вопросы радиоэлектроники. 2013. Т. 3. № 2. С. 76–83.
- Марков А.С., Цирлов В.Л. Опыт выявления уязвимостей в зарубежных программных продуктах // Вопросы кибербезопасности. 2013. № 1 (1). С. 42–48.

DOI: 10.15827/0236-235X.112.166-174

Received 28.09.15

A METHODOLOGICAL FRAMEWORK OF ANALYSIS AND SYNTHESIS OF SECURE SOFTWARE DEVELOPMENT CONTROLS

Barabanov A.V., Ph.D. (Engineering), Head of Department, ab@cnpo.ru

(Research And Production Association "Echelon", Elektrozavodskaya St. 24, Moscow, 107023, Russian Federation);

Markov A.S., Dr.Sc. (Engineering), Senior Researcher, Professor, a.markov@bmstu.ru;

Tsirlov V.L., Ph.D. (Engineering), Associate Professor, v.tsirlov@bmstu.ru

(Bauman Moscow State Technical University, 2nd Baumanskaya St. 5, Moscow, 105005, Russian Federation)

Abstract. The paper considers important issues of secure software products standardization of serial production. The authors analyze organizational and technical measures aimed to reduce the number of vulnerabilities when developing and maintaining secure automated systems software. They also systematize standards and guidelines for secure software development. The authors analyze the applicability of the existing methodological approaches to secure software development during information security assessment (including for the software certification). They show harmonization expediency of the developed regulations and practical measures with the international standards like ISO 15408 and ISO 12207. The paper introduces the concept of secure software. There also is a basic set of requirements to assess compliance of the secure software development process. In this case the set of requirements should be based on information security policies and current threats. There is an example of developed requirements. The authors developed the original conceptual model for the analysis and synthesis of secure software development controls based on a set of generated requirements. The conceptual model gives software developers the ability of science-based choice of software development methods. The authors suggest a general method of selecting a secure software development set. There is the indirect evidence of the proposed approach effectiveness. It is noted that the proposed approach was the basis for a national standard regarding security software development.

Keywords: information security, software security, secure lifecycle, software development security, secure software engineering, vulnerability, common criteria, application security controls.

References

1. Howard M., Lipner S. *The Security Development Lifecycle: A Process for Developing Demonstrably More Secure Software*. Microsoft Press, 2006, 352 p.
2. Markov A., Luchin D., Rautkin Y., Tsirlov V. Evolution of a Radio Telecommunication Hardware-Software Certification Paradigm in Accordance with Information Security Requirements. *Proc. of the 11th Int. Siberian Conf. on Control and Communications (SIBCON-2015)*. IEEE, 2015, pp. 1–4.
3. Lipaev V.V. Software Engineering basic standards development. *Programmnaya inzheneriya* [Software Engineering]. 2012, no. 6, pp. 2–7 (in Russ.).
4. Biró M., Molnár B. Synergies Between the Common Criteria and Process Improvement. *LNCS*. 2007, no. 4764, pp. 31–45.
5. Kara M. Review on Common Criteria as a Secure Software Development Model. *IJCSIT*. 2012, vol. 4, no. 2, pp. 83–94.
6. Lee M.-G., Sohn H.-J., Kim B.-M., Kim J. B. A Study on Secure SDLC Specialized in Common Criteria. *ASTL*. 2015, no. 93, pp. 11–23.
7. Barabanov A., Markov A., Fadin A., Tsirlov V., Shakhlov I. Synthesis of Secure Software Development Controls. *Proc. of the 8th Int. Conf. on Security of Information and Networks (SIN '15)*. ACM New York, NY, USA, 2015, pp. 93–97.
8. Barabanov A.V. The standardization of the process of developing security software. *Voprosy kiberbezopasnosti* [Cybersecurity Issues]. 2013, no. 1 (1), pp. 37–41 (in Russ.).
9. De Win B., Scandariato R., Buyens K., Grégoire J., Joosen W. On the Secure Software Development Process: CLASP, SDL and Touchpoints Compared. *Information and Software Technology*. 2009, vol. 51, no. 7, pp. 1152–1171.
10. Essafi M., Ghezala H.B. Meta-Modeling Based Secure Software Development Processes. *IJSSE*. 2014, no. 5 (3), pp. 56–74.
11. Fal' A. M. Standardization in information security management. *Cybernetics and Systems Analysis*. 2010, vol. 46, no. 3, pp. 512–515.
12. Barabanov A., Markov A. Modern Trends in The Regulatory Framework of the Information Security Compliance Assessment in Russia Based on Common Criteria. *Proc. of the 8th Int. Conf. on Security of Information and Networks (SIN '15)*. ACM New York, NY, USA, 2015, pp. 30–33.
13. Nevlyudov I.Sh., Andrushevich A.A., Evseev V.V. Software development lifecycle analysis for corporate information systems. *Vostochno-Evropeysky zhurnal peredovykh technology* [Eastern-European Journal of Enterprise Technologies]. 2010, vol. 6, no. 8 (48), pp. 25–27.
14. Riber G., Malmkvist K., Shcherbakov A. Mapping the application security terrain. *Voprosy kiberbezopasnosti* [Cybersecurity Issues]. 2014, no. 1 (2), pp. 36–39 (in Russ.).
15. Viega J., McGraw G. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley Prof. Publ., 2001, 528 p.
16. Avetisyan A.I., Belevantsev A.A., Chuklyaev I.I. The technologies of static and dynamic analyses detecting vulnerabilities of software. *Voprosy kiberbezopasnosti* [Cybersecurity Issues]. 2014, no. 3 (4), pp. 20–28 (in Russ.).
17. Demin A.A., Karpunin A.A., Ganev Yu.M. Verification and validation methods for complex software systems. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 4 (108), pp. 229–233 (in Russ.).
18. Ardi S., Shahmehri N. Introducing Vulnerability Awareness to Common Criteria's Security Targets. *Proc. of 4th Int. Conf. on Software Engineering Advances. ICSEA '09. IEEE Xplore Digital Library*. 2009, pp. 419–424.
19. Markov A.S., Lariontseva E.A., Stelmashuk N.N. Multifactor planning models for information security software certification tests. *Voprosy radioelektroniki* [Radio Electronics Questions]. 2013, vol. 3, no. 2, pp. 76–83 (in Russ.).
20. Markov A.S., Tsirlov V.L. Experience in identifying vulnerabilities in software. *Voprosy kiberbezopasnosti* [Cybersecurity Issues]. 2013, no. 1 (1), pp. 42–48 (in Russ.).

ЭКВАЦИОНАЛЬНАЯ ХАРАКТЕРИСТИКА ФОРМУЛ LTL*Ю.П. Кораблин, д.т.н., профессор, y.p.k@mail.ru;**А.А. Шипов, аспирант, a-j-a-1@yandex.ru**(Российский государственный социальный университет,
ул. Вильгельма Пика, 4, г. Москва, 129256, Россия)*

Программные системы с каждым днем становятся все более сложными и комплексными, поэтому требуются такие инструменты, которые позволяли бы относительно легко выполнять проверку их работы на соответствие заданным спецификациям, особенно, когда речь идет о больших и распределенных программных системах.

Для описания проверяемых условий верифицируемых моделей сегодня используются такие механизмы, как логика линейного времени LTL и логика ветвящегося времени CTL. Однако, как показывает практика, с помощью данных механизмов можно сформулировать лишь относительно небольшое множество однотипных условий, что может существенно усложнить процесс верификации или же сделать его вовсе неэффективным для модели конкретной системы.

Проблема корректной формулировки проверяемых на модели свойств является одной из ключевых, так как от этого будет зависеть весь процесс верификации. Таким образом, наличие мощных инструментов и методов, позволяющих однозначно формулировать широкий класс проверяемых свойств, является необходимым требованием для достижения наилучших результатов.

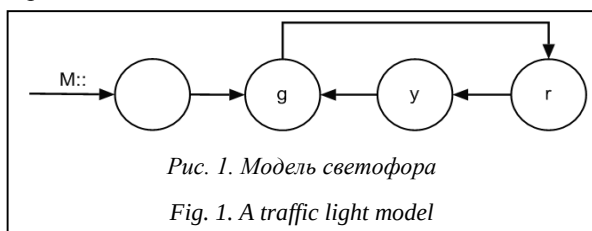
В статье предложен механизм, использование которого позволяет существенно расширить группу условий, формулируемых по отношению к проверяемым моделям. Данный эффект достигается путем расширения выразительности логики линейного времени LTL с помощью предлагаемого в статье метода, что в итоге позволяет увеличить эффективность процесса верификации.

Теоретический материал статьи подкреплен рядом наглядных примеров работы данного метода, демонстрирующих его практичность. Также приведен пример верификации свойств, сформулированных на основе предложенного метода, для конкретной модели.

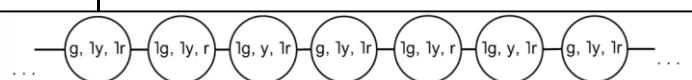
Ключевые слова: верификация, модель Крипке, автомат Бюхи, формула временной логики, LTL, CTL.

Прежде чем выполнять верификацию для конкретной системы, необходимо сформулировать те ее свойства, наличие или отсутствие которых требуется доказать. Привычная разговорная и письменная речь зачастую не позволяет сделать это однозначно, поэтому были разработаны специальные механизмы для решения данной проблемы [1, 2], в частности, метод, основанный на использовании различных моделей временной логики: логики линейного времени (LTL), ветвящегося времени (CTL), а также обобщающей их логики CTL* [3–5]. Данные логики лишены недостатков с точки зрения однозначности формулируемых на их базе свойств. Однако, как показывает практика, их мощность позволяет формулировать лишь относительно небольшое количество однотипных условий, а этого, в свою очередь, может быть недостаточно для проверки тех или иных свойств модели конкретной системы.

Рассмотрим, например, модель работы светофора, представленную на рисунке 1 структурой Крипке [6].



Предикаты модели принимают следующие значения: g – включен зеленый сигнал светофора, r – красный, y – желтый. Как видим, светофор из своего некоторого первоначального состояния переходит в режим, в котором наступает строгое чередование зеленого, красного и желтого сигналов. Предположим, что стоит задача верификации работы этого режима с доказательством строгого чередования сигналов. Представить это можно в виде развертки, изображенной на рисунке 2.



И здесь встает проблема, решение которой невозможно в рамках любой из имеющихся логик: определить свойство строгого чередования трех предикатов на бесконечном интервале в случае необходимости его проверки и сформулировать чередование любого произвольного конечного числа предикатов. В общем виде в LTL свойство чередования для предложенного светофора можно определить с помощью оператора X (на следующем шаге) как $\varphi = Xg \wedge XXr \wedge XXXy \wedge XXXXg \wedge \dots$. Несмотря на то, что запись $G(g \wedge Xr \wedge XXy)$ кажется интуитивно правильной для данного примера, можно показать, что в действительности она описывает последовательность совершенно иного ха-

рактера. А именно, используя правило $G(p \wedge q) = G(p) \wedge G(q)$, можно представить данную запись в виде $G(g) \wedge G(Xr) \wedge G(XXy)$. Применяя правило рекурсивного определения оператора G вида $Gr = p \wedge X(r) \wedge XX(r) \wedge XXX(r) \dots$ для каждого из конъюнктов, получим

$$Gg = g \wedge X(g) \wedge XX(g) \wedge XXX(g) \dots$$

$$G(Xr) = Xr \wedge X(Xr) \wedge XX(Xr) \wedge XXX(Xr) \dots$$

$$G(XXy) = XXy \wedge X(XXy) \wedge XX(XXy) \wedge XXX(XXy) \dots$$

Объединив их, получим последовательность вида $\varphi = g \wedge X(g \wedge r) \wedge XX(g \wedge r \wedge y) \wedge XXX(g \wedge r \wedge y) \dots$

Очевидно, что данная последовательность не совпадает с описанной ранее последовательностью. Этот пример иллюстрирует неэффективность оператора G в задачах определения строгого чередования предикатов на бесконечной последовательности состояний.

Таким образом, встает проблема разработки некоторого механизма формулировки свойств системы, который позволял бы расширить базовое множество типовых выражений и был бы достаточно удобным для выполнения проверки на моделях.

Эквационная характеристика операторов LTL (RLTL)

Рассмотрим рекурсивные представления основных базовых операторов логики LTL (табл. 1).

Таблица 1

Рекурсивные представления операторов LTL

Table 1

LTL operators' recursive presentations

LTL	RLTL
$F\varphi = \varphi \vee XF\varphi$	$F\varphi = \varphi \vee \Delta \circ F\varphi$
$G\varphi = \varphi \wedge XG\varphi$	$G\varphi = \varphi \circ G\varphi$
$\varphi_1 U \varphi_2 = \varphi_2 \vee \varphi_1 \wedge X(\varphi_1 U \varphi_2)$	$U(\varphi_1, \varphi_2) = \varphi_2 \vee \varphi_1 \circ U(\varphi_1, \varphi_2)$
$\varphi_1 \wedge X\varphi_2$	$\varphi_1 \circ \varphi_2$
$X\varphi$	$\Delta \circ \varphi$

Символ « $\circ$ » обозначает оператор продолжения (конкатенация выражений), что позволяет использовать оператор X в неявном виде, а также упростить его восприятие, в частности, запись $\varphi_1 \circ \varphi_2$ будет пониматься как « φ_2 следует за φ_1 ». Под символом « Δ », далее именуемым предикатом неопределенности, будем понимать неопределенное множество предикатов для конкретного состояния вычислительной последовательности относительно тех предикатов, которые принимают в нем истинностные значения.

В таблице 1 представлены все операторы, используемые в LTL. Поскольку все основные операторы LTL могут быть приведены к своим рекурсивным представлениям, авторы полагают, что приведение их к некоторому унифицированному виду, который в последующем именуется эквационной характеристикой, позволит существенно расширить описательную способность логики LTL и упростит процесс верификации ее формул.

Сочетания операторов LTL также могут быть выведены с помощью аксиом в рамках эквационной характеристики. В таблице 2 приведены некоторые сочетания операторов LTL.

Таблица 2

Сочетания операторов LTL в эквациональной форме

Table 2

Equational combinations of LTL operators

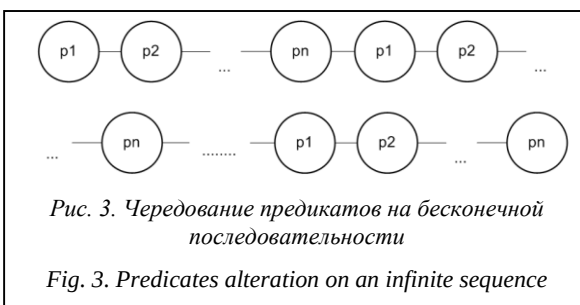
LTL	RLTL
$FG\varphi$	$F' = \varphi \circ G' \vee \Delta \circ F'$ $G' = \varphi \circ G'$
$GF\varphi$	$G' = \varphi \circ G' \vee \Delta \circ F'$ $F' = \varphi \circ G' \vee \Delta \circ F'$

Сформулируем основные аксиомы логики RLTL:

A1.	$FF\varphi = F\varphi$	(FF)
A2.	$GG\varphi = G\varphi$	(GG)
A3.	$!F\varphi = G!F\varphi$	(!F)
A4.	$!G\varphi = F!G\varphi$	(!G)
A5.	$G\varphi = \varphi \circ G\varphi$	(G)
A6.	$F\varphi = \varphi \vee \Delta \circ F\varphi$	(F)
A7.	$G\varphi \vee !G\varphi = \Delta$	($\vee G$)
A8.	$F\varphi \vee !F\varphi = \Delta$	($\vee F$)
A9.	$G\varphi_1 \circ \varphi_2 = G\varphi_1$	($G \circ$)
A10.	$F\varphi_1 \circ \varphi_2 = F(\varphi_1 \circ \varphi_2)$	($F \circ$)
A11.	$U(\varphi_1, \varphi_2) = \varphi_2 \vee \varphi_1 \circ U(\varphi_1, \varphi_2)$	(U)
A12.	$!(\varphi \circ \varphi_1) = !\varphi \vee \Delta \circ !\varphi_1$	(! $\varphi \circ$)
A13.	$! \Delta = \Delta$	(! Δ)
A14.	$\Delta \wedge \varphi = \varphi$	($\Delta \wedge$)
A15.	$\varphi = \varphi \circ \Delta$	($\circ \Delta$)

Примеры описания свойств в RLTL-нотации

Рассмотрим некоторые типы свойств, описать которые можно, лишь используя предлагаемый механизм эквационной характеристики для логики LTL. Одним из таких типов будет тип строго чередующихся последовательностей предикатов на бесконечной вычислительной последовательности состояний. В общем виде данный тип представлен на рисунке 3.



Общий вид данного типа задается системой рекурсивных уравнений Φ :

$$F_{r1} = p_1 \circ F_{r2},$$

$$F_{r2} = p_2 \circ F_{r3},$$

$$\dots$$

$$F_{rn} = p_n \circ F_{r1}.$$

Пример данного типа уже был рассмотрен при описании модели работы светофора. Сформулируем заданное для нее в общем виде условие $\varphi = Xg \wedge XXg \wedge XXXu \wedge XXXXg \wedge \dots$ с помощью эквивалентной характеристики:

$$F_{r1} = \emptyset \circ F_{r2};$$

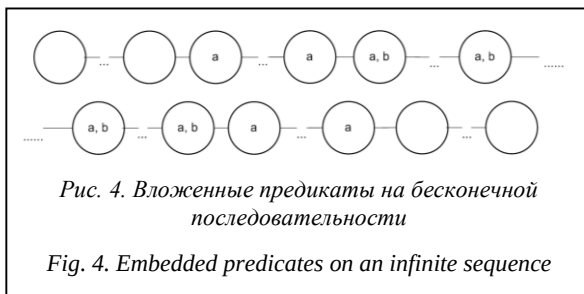
$$F_{r2} = g \circ F_{r3};$$

$$F_{r3} = r \circ F_{r4};$$

$$F_{r4} = y \circ F_{r2}.$$

Под символом « $\emptyset$ » в дальнейшем будем понимать пустое множество предикатов, означающее, что ни один из предикатов множества AP [6] не принимает значение «истина». Таким образом, сформулированное условие представляет собой систему из четырех рекурсивных уравнений, строго задающих необходимую последовательность предикатов (сигналов светофора).

Еще одним типом свойств, которые нельзя выразить стандартными формулами LTL, является вхождение (появление) вложенных последовательностей предикатов на бесконечной вычислительной последовательности состояний. В общем виде данный тип изображен на рисунке 4.



Условно данный тип может быть описан как тип постепенного переключения режимов работы системы, где переход в каждый последующий режим возможен лишь в случае корректного выполнения предыдущего, причем последовательности включения и выключения этих режимов строго взаимно обратные.

Представим данный тип в виде системы рекурсивных уравнений Φ :

$$F_{r1} = \emptyset \circ F_{r1} \vee a \circ F_{r2};$$

$$F_{r2} = a \circ F_{r2} \vee (a \wedge b) \circ F_{r3};$$

...

$$F_{r(k)} = (a \wedge b) \circ F_{r(k)} \vee a \circ F_{r(k+1)};$$

$$F_{r(k+1)} = a \circ F_{r(k+1)} \vee \emptyset \circ F_{r1}.$$

В качестве примера можно рассмотреть модель работы подсистемы космического спутника, отвечающую за работу передатчика и приемника радиосигнала (рис. 5).

Таким образом, развертка данной модели будет выглядеть так, как показано на рисунке 6.

Система рекурсивных уравнений, характеризующая данную

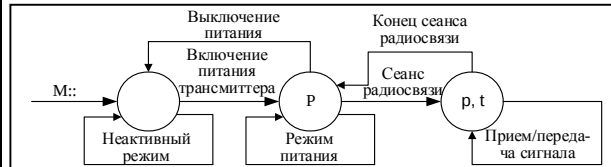


Рис. 5. Модель работы передатчика спутника

Fig. 5. A satellite transmitter operating model

последовательность, будет состоять из четырех уравнений:

$$F_{r1} = \emptyset \circ F_{r1} \vee p \circ F_{r2};$$

$$F_{r2} = p \circ F_{r2} \vee (p \wedge t) \circ F_{r3};$$

$$F_{r3} = (p \wedge t) \circ F_{r3} \vee p \circ F_{r4};$$

$$F_{r4} = p \circ F_{r4} \vee \emptyset \circ F_{r1}.$$

Верификация RLTL

Покажем, что, задавая некоторое свойство, представленное системой рекурсивных уравнений, легко выполнить его верификацию на заданной модели. В качестве примера возьмем рассмотренную выше модель светофора, представленную на рисунке 1, а в качестве проверяемого условия – систему рекурсивных уравнений Φ , означающую то, что «со второго состояния светофора начинается строгое чередование зеленого, красного и желтого сигналов»:

$$\Phi = F_{r1} = \emptyset \circ F_{r2};$$

$$F_{r2} = g \circ F_{r3};$$

$$F_{r3} = r \circ F_{r4};$$

$$F_{r4} = y \circ F_{r2}.$$

Для выполнения верификации формул RLTL будет использована та же самая методика верификации, что и для формул LTL, описанная в [6].

На первом этапе сформулируем отрицание проверяемого условия, используя аксиомы, приведенные выше:

$$\neg \Phi = \neg F_{r1} = \neg \emptyset \vee \Delta \circ \neg F_{r2};$$

$$\neg F_{r2} = \neg g \vee \Delta \circ \neg F_{r3};$$

$$\neg F_{r3} = \neg r \vee \Delta \circ \neg F_{r5};$$

$$\neg F_{r4} = \neg y \vee \Delta \circ \neg F_{r2}.$$

На втором этапе для полученной системы рекурсивных уравнений построим контрольный автомат. Для упрощения процесса построения контрольного автомата применим аксиому A15 к полу-

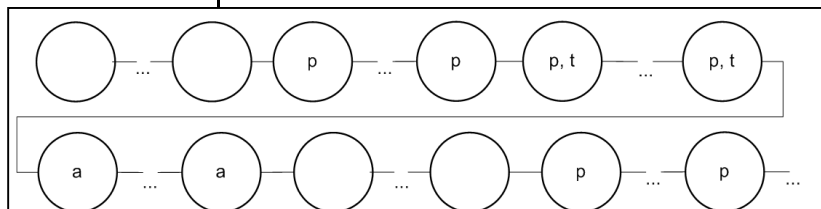


Рис. 6. Развертка модели работы передатчика спутника

Fig. 6. A developed view of a satellite transmitter operating model

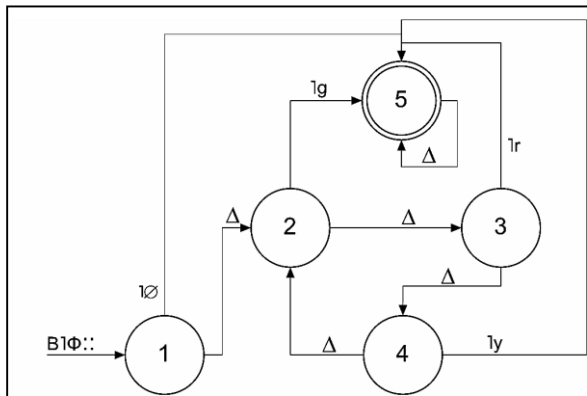


Рис. 7. Контрольный автомат Бюхи B1Φ

Fig. 7. Check Büchi automata B1Φ

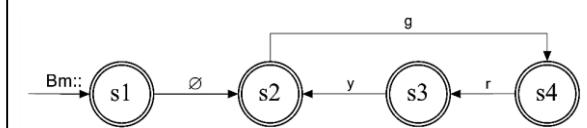


Рис. 8. Автомат Бюхи Bm для модели светофора

Fig. 8. Büchi automata Bm for a traffic light model

ченной системе рекурсивных уравнений и получим следующую систему рекурсивных уравнений:

$$1\Phi = F_{r1} = 1\emptyset \circ F_{r5} \vee \Delta \circ F_{r2};$$

$$F_{r2} = 1g \circ F_{r5} \vee \Delta \circ F_{r3};$$

$$F_{r3} = 1r \circ F_{r5} \vee \Delta \circ F_{r5};$$

$$F_{r4} = 1y \circ F_{r5} \vee \Delta \circ F_{r2};$$

$$F_{r5} = \Delta \circ F_{r5}.$$

На третьем этапе необходимо представить 1Φ в виде автомата Бюхи B1Φ. Для этого построим такой автомат, в котором состояниям будут сопоставлены переменные F_i нашей рекурсивной системы уравнений, а переходам – предикаты. Как видно из рисунка 7, сделать это достаточно просто. Допускающими состояниями являются те, в которые существует хотя бы один переход по предикату, отличному от предиката неопределенности Δ .

Теперь построим композицию автомата B1Φ и автомата Bm нашей модели светофора (рис. 8) согласно методу, описанному в [6]. Результатом композиции перехода, помеченного предикатом неопределенности Δ , с переходом, помеченным некоторым предикатом ϕ , является переход, помеченный ϕ , так как неопределенность устраняется. Результат композиции представлен на рисунке 9.

Жирными стрелками показаны переходы, которые достижимы из начального состояния ($s1, 1$). Очевидно, что из

начального состояния невозможно попасть ни в одно из допускающих состояний, что означает невыполнимость 1Φ , а следовательно, выполнимость Φ . Таким образом, свойство «со второго состояния светофора начинается строгое чередование зеленого, красного и желтого сигналов» действительно выполняется.

В заключение отметим, что в статье была рассмотрена проблема формулирования сложных верифицируемых свойств. Как показала практика, LTL и CTL не позволяют формулировать большой спектр условий, среди которых могут оказаться именно те, что необходимы для верификации конкретной модели. Предложенный в работе механизм эквивалентной характеристики для логики LTL позволяет существенно расширить возможности ее описательных свойств. На основе данного механизма были рассмотрены два конкретных примера, относящихся к конкретным типам вычислительных последовательностей, представление которых невозможно в рамках стандартных формул LTL. Для первого примера продемонстрирован алгоритм верификации, показавший, что процесс построения контрольного автомата на базе системы рекурсивных уравнений является гораздо более удобным, интуитивно понятным и наглядным.

Авторы полагают, что использование данного представления формул временной логики позволит существенно увеличить гибкость и практичность верификации, поскольку предлагаемый метод может быть с легкостью внедрен в уже существую-

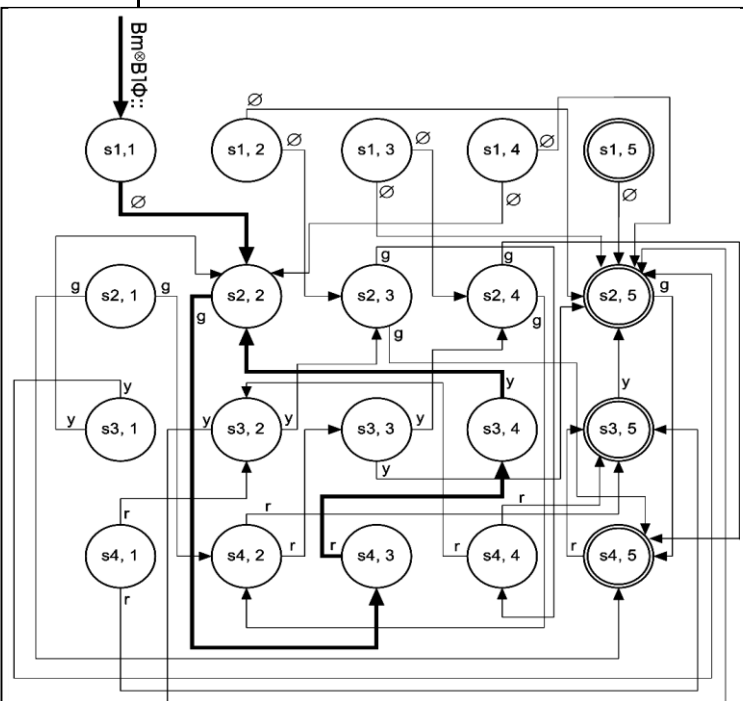


Рис. 9. Композиция исходного и контрольного автоматов

Fig. 9. The composition of original and check automata

щие технологии верификации на базе метода Model Checking или проверки на моделях.

Литература

1. Кораблин Ю.П. Семантика языков распределенного программирования: учеб. пособие. М.: Изд-во МЭИ, 1996. 102 с.
2. Hopcroft J.E., Motwani R., Ullman J.D. *Introduction to Automata Theory, Languages, and Computation*, 3/E – Pearson/Addison Wesley, 2007, 535 p.
3. Велдер С.А., Лукин М.А., Шалута А.А., Яминов Б.Р. Верификация автоматных программ. СПб: Наука, 2011. 244 с.
4. Kröger F., Merz S. *Temporal Logic and State Systems*. Springer (March 27, 2008), 436 p.
5. Manna Z., Pnueli A. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. 1992, 427 p.
6. Карпов Ю.Г. *Model Checking. Верификация параллельных и распределенных программных систем*. СПб: БХВ-Петербург, 2010. 552 с.
7. Кораблин Ю.П., Куликова Н.Л. Логические методы доказательства и тестирования программ // Информационные средства и технологии: тез. докл. Междунар. конф. М.: Изд-во МФИ, 1996. 9 с.
8. Кораблин Ю.П. Семантика языков программирования. М.: Изд-во МЭИ, 1992. 100 с.
9. Кораблин Ю.П., Косакян М.Л. Анализ моделей программ на языке асинхронных функциональных схем средствами темпоральной логики линейного времени // Программные продукты и системы. 2014. № 2 (106). С. 5–10.
10. Алагич С., Арbib М. Проектирование корректных структурированных программ. М.: Радио и связь, 1979. 292 с.
11. Hoare C.A.R. *An axiomatic basis for computer programming*. Northern Ireland: Queen's Univ. of Belfast, 1969, pp. 576–580.
12. Ábrahám E., Havelund K. (Eds). *Tools and Algorithms for the Construction and Analysis of Systems*. Springer (March 24, 2014), 652 p.
13. Bérard B., Bidoit M., Finkel A., Laroussinie F., Petit A., Petrucci L., Schnoebelen P. *Systems and Software Verification. Model Checking Techniques and Tools*. Springer, 2001, 190 p.

DOI: 10.15827/0236-235X.112.175-179

Received 22.07.15

EQUATIONAL CHARACTERISTICS OF LTL FORMULAS

Korablin Yu.P., Dr.Sc. (Engineering), Professor, y.p.k@mail.ru;

Shipov A.A., Postgraduate Student, a-j-a-1@yandex.ru

(Russian Federation State Social University, Vilgelma Pika St. 4, Moscow, 129256, Russian Federation)

Abstract. Day by day software systems are becoming more and more complex. Therefore, we need to have some useful instruments to check their capability according to specifications, especially large distributed software systems.

Nowadays, to describe the verifying model conditions it is common to use such mechanisms as linear temporal logic (LTL) and computational tree logic (CTL). However, experience has proven that these mechanisms can help formulate only a relatively small set of the same-type conditions. It can complicate the verification process or make it ineffective for a particular system model.

Correct formulation of model's verifying properties is the key problem, as the whole verification process depends on it. Thus, to achieve best results it is required to use powerful tools and techniques, which clearly formulate a wide class of verifying properties.

The article describes a mechanism that can significantly extend the group of formulated conditions according to verifiable models. This effect can be achieved by expanding LTL expressivity using the proposed method, which increases the efficiency of the verification process.

The article provides a set of illustrative examples of the method, which demonstrates its practical application. The article also provides the example of properties verification for a particular model based on the proposed method.

Keywords: verification, Kripke structure, Büchi automaton, temporal logic formula, LTL, CTL.

References

1. Korablin Yu.P. *Semantika yazykov raspredelenogo programmirovaniya* [Distributed Programming Language Semantics]. Study guide. V.P. Kutepov (Ed.). Moscow, MEI Publ., 1996, 102 p.
2. Hopcroft J.E., Motwani R., Ullman J.D. *Introduction to Automata Theory, Languages, and Computation*. 3/E. Pearson/Addison Wesley Publ., 2007, 535 p.
3. Velder S.A., Lukin M.A., Shaluta A.A., Yaminov B.R. *Verifikatsiya avtomatnykh programm* [Verification of automata-based programs]. St.-Petersburg, Nauka Publ., 2011, 244 p.
4. Kröger F., Merz S. *Temporal Logic and State Systems*. Springer Publ., 2008, 436 p.
5. Manna Z., Pnueli A. *The Temporal Logic of Reactive and Concurrent Systems: Specification*. 1992, 427 p.
6. Karpov Yu.G. *Model Checking. Verifikatsiya parallelnykh i raspredelennykh programmykh sistem* [Model Checking. Verification of parallel and distributed program systems]. BHV-Peterburg Publ., 2010, 552 p.
7. Korablin Yu.P., Kulikova N.L. *Logical Proof Methods and Testing Programs. Tezisy, доклад. Mezhdunar. konf. "Informatsionnye sredstva i tekhnologii"* [Proc. of the Int. Conf. "Information means and technologies"]. Moscow, 1996, 9 p.
8. Korablin Yu.P. *Semantika yazykov programmirovaniya* [Programming Language Semantics]. Moscow, MEI Publ., 1992, 100 p.
9. Korablin Yu.P., Kosakyan M.L. Analysis of the asynchronous functional schemes program models using linear-time temporal logic. *Programmye produkty i sistemy* [Software & Systems]. 2014, no. 2 (106), pp. 5–10 (in Russ.).
10. Alagich S., Arbib M. *Proektirovanie korrektnykh strukturirovannykh programm* [Valid Structured Programs Design]. Moscow, Radio i svyaz Publ., 1979, 292 p.
11. Hoare C.A.R. *An Axiomatic Basis for Computer Programming*. N.Ireland, Queen's Univ. of Belfast Publ., 1969, pp. 576–580.
12. Ábrahám E., Havelund K. *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Publ., 2014, 652 p.
13. Bérard B., Bidoit M., Finkel A., Laroussinie F., Petit A., Petrucci L., Schnoebelen P. *Systems and Software Verification. Model Checking Techniques and Tools*. Springer Publ., 2001, 190 p.

УДК 004.94

Дата подачи статьи: 15.06.15

DOI: 10.15827/0236-235X.112.180-187

ОСНОВНЫЕ ПОНЯТИЯ ФОРМАЛЬНОЙ МОДЕЛИ СЕМАНТИЧЕСКИХ БИБЛИОТЕК И ФОРМАЛИЗАЦИЯ ПРОЦЕССОВ ИНТЕГРАЦИИ В НЕЙ

О.М. Атаева, младший научный сотрудник, oli@ultimeta.ru;

В.А. Серебряков, д.ф.-м.н., профессор, зав. отделом, serebr@ultimeta.ru

(Вычислительный центр им. А. А. Дородницына РАН, ул. Вавилова, 40, г. Москва, 119333, Россия)

Развитие современных технологий подталкивает к переопределению понятия контента библиотеки, в качестве которого могут выступать и традиционные описания печатных изданий, и любые другие типы объектов. При этом контент цифровых библиотек и физические объекты могут быть связаны различными способами. В работе рассматриваются библиотеки как хранилища структурированных разнообразных данных с возможностью их интеграции с другими источниками данных. Приведены структура тезауруса для возможности определения их тематической направленности и основные понятия, необходимые для описания таких библиотек. Определяя такие понятия, как информационные ресурсы, наборы атрибутов, информационные объекты и другие, связанные с ними, формируют понятийную основу для некоторой предметной области создаваемой семантической библиотеки. Тезаурус же, в свою очередь, обеспечивает терминологическую поддержку этих понятий, облегчая навигацию по информационным объектам системы, поддерживает процесс уточнения и расширения запросов пользователей к системе. Также в статье описаны понятия, необходимые для детализации работ по интеграции данных, основной упор делается на понятия, используемые в процессе приведения данных. Важной характеристикой любого набора данных независимо от его структуры является понятие качества данных. Опираясь на оценку качества данных, можно давать объективную оценку эффективности процессов, происходящих в семантической библиотеке, важнейшим из которых является интеграция данных с другими источниками. Формальная модель понятий, описанная в данной части работы, используется в дальнейшем для описания онтологии такой библиотеки.

Ключевые слова: *семантические библиотеки, качество данных, приведение данных, модель данных.*

Цифровые библиотеки воспринимаются обычными пользователями как электронные версии каталогов традиционных библиотек, которые содержат описания физических объектов библиотеки, обычно книг или других печатных изданий. Определения тематики, содержания и структуры объектов рассматриваются и воспринимаются ими как дополнительные, но необязательные функции таких библиотек.

Развитие Интернета и семантических технологий вносит свои коррективы [1], позволяет более широко взглянуть на понятие цифровых библиотек и обобщить накопившийся опыт реализации информационных систем в разных областях знаний для библиотек. Само понятие библиотеки в контексте Интернета [2] приобретает совершенно другой смысл, обозначает, как минимум, активное вовлечение пользователя в процессы, предлагаемые библиотеками, и предполагает участие пользователей в процессе создания, поиска и классификации того контента библиотеки, который необходим этому конкретному пользователю.

Развитие современных технологий также подталкивает к переопределению понятия контента библиотеки, в качестве которого могут выступать не только традиционные описания печатных изданий, но и любые другие типы объектов [3], например мультимедийные: видео, звук, фотографии, музейные экспонаты, коллекции минералов, архивные материалы и многое другое.

При этом необходимо четко понимать, что контент цифровых библиотек и физические объекты

могут быть связаны различными способами. Это происходит из-за того, что физически объект существует в реальном мире в одном экземпляре, а в цифровой библиотеке всего лишь используется его описание. При этом описаний может быть несколько, они могут быть различными по структуре и смыслу, и каждое описание, ссылаясь на реальный объект, имеет собственный уникальный идентификатор, который позволяет идентифицировать конкретное описание объекта со ссылкой на него в реальном мире. Фактически цифровой ресурс может определяться как конгломерат разных описаний одного реального ресурса, представляя его общую объединенную модель данных.

В этом смысле Интернет также может рассматриваться как библиотека, стихийно наполняемая пользователями без видимого порядка и структуры. Каждая страница имеет собственный идентификатор в виде URL и относится к некоторому объекту реального мира, и таких страниц могут быть тысячи. Чаще всего автоматическая обработка таких страниц для выявления полезной информации и взаимосвязей является сложной задачей и предполагает большие трудозатраты на извлечение реально нужной информации по узкой предметной области.

Предпосылки работы

Для облегчения процесса автоматической обработки информации в сети некоторое время назад была предложена концепция Linked Open Data (да-

лее LOD) [4] для размещения и описания данных, опирающаяся на уже имеющиеся наработки парадигмы Semantic Web [5]. Основные принципы LOD призывают использовать для идентификации реальных ресурсов, их цифровых описаний и взаимосвязей уникальные идентификаторы URI. Причем эти URI должны обеспечивать доступ к описаниям объектов по протоколу HTTP, представлять описание в виде RDF [6], которое дает возможность автоматической обработки информации, и содержать в своем описании ссылки в виде URI на другие взаимосвязанные ресурсы и их описания. При соблюдении этих принципов обеспечивается стандартизированный механизм доступа к данным и поддерживается глобальный обмен данными независимо от модели данных конкретного узла, предоставляя возможность их интеграции, основываясь на URI.

Парадигма Semantic Web позволяет структурировать описания ресурсов и представлять их в виде RDF, основываясь на онтологиях [7]. Онтология любой предметной области определяет ее понятия, тип, структуру, совокупность словарей и классификаторов, которые представляют тезаурус предметной области, обеспечивает доступ к знаниям предметной области в разных источниках. Онтологии позволяют вырабатывать и фиксировать общее понимание области знания, представлять знания в виде, удобном для их автоматизированной обработки, обеспечивать возможность получения и накопления новых знаний, а также возможность их многократного использования. Тезаурус же обеспечивает терминологическую поддержку предметной области, облегчает навигацию по информационным объектам системы, поддерживает процесс уточнения и расширения запросов пользователей к системе.

Все это послужило предпосылками для лавинообразного роста источников данных, интегрированных в LOD, и традиционные библиотеки, являясь по сути центрами данных, с одной стороны, не могли игнорировать этот процесс, с другой – не только получили импульс к развитию и обогащению своих данных, но и стали трансформироваться и превращаться в хранилища ресурсов самых разнообразных типов, поддерживая их взаимосвязи с другими источниками из LOD. Основная проблема, возникающая в связи с этим, – гетерогенность онтологий разных источников, которая может препятствовать связыванию данных [8]. Однако существует множество исследований, с разным успехом преодолевающих эту проблему. В частности, одним из распространенных способов является активное использование и анализ связей как на уровне источников, так и на уровне самих данных.

Таким образом, авторы статьи рассматривают библиотеки как хранилища структурированных разнообразных данных с возможностями их интеграции в облако LOD и определения их тематической направленности. При этом подразумевается,

что поддерживается вся традиционная для электронных библиотек функциональность: создание, редактирование, поиск, идентификация ресурсов. В дальнейших работах планируется представить онтологию такой библиотеки, архитектуру приложения и описать реализацию прототипа этого приложения на примере данных библиотеки «Научное наследие России» [9]. При этом пока опускается описание процессов взаимодействия пользователя с этой библиотекой, а основное внимание уделяется ее основным понятиям, которые необходимы для описания модели данных и процессов поступления (обогащения) и поиска данных для цифровой библиотеки определенной тематики, интегрированной с различными источниками данных.

Основные понятия

Семантические библиотеки активно развиваются на протяжении последнего десятилетия [10], но их понятийная структура еще не устоялась. Концептуальная модель электронных библиотек с определениями важнейших представлений об архитектуре, ресурсах и их функциональности была определена в программном документе DELOS (Digital Library Reference Model) [11]. С другой стороны, эффективность исследований в этой области зависит от стандартизации и формализации собственно описаний ресурсов таких библиотек и процессов их представления.

Остановимся на определениях для построения формальной модели любых типов ресурсов рассматриваемой семантической библиотеки, которые лежат в основе ее построения. Фактически понятия делятся на две категории: первая включает определения понятий контента семантической библиотеки, а вторая – определения понятий, необходимых для определения процессов интеграции контента этих ресурсов (здесь под интеграцией подразумевается весь спектр задач, например, определение отображения разных наборов атрибутов одного ресурса, отображение ресурсов источника на ресурсы семантической библиотеки, качество данных, процесс «очистки данных» с выявлением дубликатов, например, связывание с источником данных, классификация ресурсов средствами тезауруса предметной онтологии, распределенный поиск по источникам). На основе этих определений описываются основные процессы, например, интегрирование данных из разных источников, категоризация/классификация, выявление дубликатов на уровне данных, отображение разных моделей данных источников на заданную предметную область и т.д.

Роль тезауруса в семантической библиотеке.

Определяя информационные ресурсы, наборы атрибутов и информационные объекты, можно формировать понятийную основу конкретной предметной области для семантической библио-

теки. Тезаурус же, в свою очередь, обеспечивает терминологическую поддержку понятий, помогая решать разнообразные задачи.

Одна из таких актуальных задач – поддержка классификации ресурсов в библиотеке. Зачастую направлений классификации бывает несколько, в некоторых случаях классификаторы могут динамически меняться в процессе жизнедеятельности системы. Более того, со временем могут потребоваться новые классификаторы, которые в идеале должны безболезненно и с минимальными затратами внедряться в библиотеку, не требуя внесения изменений в систему и не затрагивая структуру данных хранилища системы. При этом часто возникают ситуации, когда классификацию необходимо применить к различным типам ресурсов, причем классифицируемые типы ресурсов могут определяться также по ходу развития системы. Мы называем такую поддержку процедуры классификации гибкой классификацией.

Основные определения для описания структуры контента библиотеки: o – информационный объект, $O_B = \{o_1, \dots, o_n\}$ – множество информационных объектов; r – информационный ресурс, $R = \{r_1, \dots, r_m\}$ – множество информационных ресурсов (типов информационных объектов); $TYPE(o)=r$ каждому объекту соответствует информационный ресурс; $IsRe(r_1, r_2)$ определяет иерархию ресурсов, то есть r_2 подресурс r_1 , ресурс может иметь 0 и более подресурсов.

Атрибуты: $A = \{a_1, \dots, a_k\}$ – множество атрибутов информационных ресурсов, $Z(a)$ – значение атрибута a .

Тип атрибутов: A_D – множество атрибутов, значением которых может являться любое значение из некоторого множества D , в котором не может быть информационных ресурсов системы:

$$\forall a_i, a_i \in A_D: Z(a_i) = d, d \in D, D \notin O_B.$$

A_I – множество атрибутов, значением которых может являться любой объект из I_O :

$$\forall a_i, a_i \in A_I: Z(a_i) = o, o \in O_B;$$

$$A_D \vee A_I = A, A_D \wedge A_I = \emptyset.$$

Вид атрибутов: $V = \{v_1, \dots, v_t\}$.

Атрибут a может быть нескольких видов:

$\forall a, a \in A: VIEW(a) = \{v_k\}, v_k \in V, 1 \leq k \leq t$, далее для простоты будем обозначать вид атрибута $VIEW(a) = v_k$.

Для атрибутов определены область значений $DOM(a)=R'$, $R' \in R$, и область определения $RAN(a)=X: X \in D$ или $X \in O_B$.

Наборы атрибутов: SA – множество таких подмножеств атрибутов, в каждое из которых входят хотя бы по одному атрибуту каждого вида, каждое такое подмножество называется набором атрибутов: $\forall v, A' v \in V, A' \in SA \exists a, a \in A' VIEW(a)=v, v \in V$.

$SET(r) = A''$ – соответствие информационного ресурса некоторым наборам атрибутов $A'' = \{A'\}$, $A' \in SA$.

Если информационному ресурсу r соответствует некоторый набор атрибутов $A' = \{a_1, \dots, a_t\}$, то все объекты $o \in O_B$, такие что $TYPE(o)=r$, представляются как набор значений этих атрибутов $(Z(a_1), \dots, Z(a_t))$.

По своим функциям атрибуты можно разделить на следующие пересекающиеся виды: идентифицирующие, озаглавливающие, обязательные, классифицирующие, поисковые, описательные. В процессе обработки данных каждая задача в системе использует определенный тип атрибутов. Например, идентифицирующими атрибутами ресурса будем называть подмножество атрибутов набора атрибутов, необходимых и достаточных для однозначного определения каждого из его информационных объектов. Классифицирующие атрибуты обеспечивают поддержку задач классификации объектов и сохранения ее результатов.

В соответствии с типом значений атрибуты делятся на однозначные, многозначные, имеющие множество однотипных значений, которые могут составлять мультимножество, множество, список, массив. Для описания формальной общей картины эти тонкости пока можно опустить, полагая, что мы можем включить в набор атрибутов столько однотипных атрибутов, сколько требуется.

Представление информационного ресурса $P(A', v)$ определяется типом используемых в нем атрибутов, соответствующих ресурсу (используется, например, для автоматической генерации описания объекта для пользователя или какого-то агента). Фактически представляет выборку значений атрибутов объекта некоторого ресурса для заданного вида атрибутов.

Определения для описания основных процессов обработки контента библиотеки.

Модель данных источника: $S_X = \{(r_x, A_r)\}$, где r_x и A_r – ресурсы и их атрибуты, представленные в этом источнике; G – функция отображения элементов модели данных источника на информационные ресурсы системы и их набор атрибутов: $G(r_x, A_r) = (r, A')$, $SET(r) = A'$, $A' \in SA$.

Источник данных: $X = (S_X, G)$.

Тезаурус предметной области: $Th = (P, T, R)$, P – набор понятий, T – набор вербальных терминов для понятий, R – набор горизонтальных и вертикальных связей между понятиями и терминами.

F – функция сопоставления каждому информационному объекту из O_B соответствующих терминов из T : $F(o_i) = \{t\}$, $t \in T$, $o_i \in O_B$.

Качество данных – это набор характеристик $Q = (q_1, \dots, q_m)$. DQ – функция, оценивающая информационные объекты на соответствие условиям характеристик качества и ставящая в соответствие некоторое число k из диапазона от 0 до 1, где значения, близкие к 1, считаются более качественными. $DQ(o_i) = k = \sum q_j(o_i)/m$; если объект o_i удовлетворяет характеристике q_j , то $q_j(o_i)=1$, в противном случае $q_j(o_i)=0$.

Основные утверждения

1) Атрибут a_i может быть только одного типа:
 $\forall a_i \in A, A_D \wedge A_I = \emptyset, \Rightarrow a_i \in A_D \vee a_i \in A_I$.

2) Множеству атрибутов информационного ресурса соответствует хотя бы один набор атрибутов:

$\forall r = \{a_1, \dots, a_k\}, r \in R, \exists A' \subseteq S : SET(r) = A', a_j \in A', 1 \leq j \leq k$.

3) Любому информационному объекту соответствует определенный набор атрибутов:

$\forall o_i \in O_B \exists A' = SET(r), A' \subseteq S : r = TYPE(o_i)$.

4) Если атрибуты информационного ресурса входят в разные наборы атрибутов, то этому ресурсу соответствуют несколько наборов атрибутов: $\forall r, r \in R; a_i \in A_1', a_j \in A_2', a_j, a_i \notin A_1' \wedge A_2', a_j \neq a_i, \exists A_1' = SET(r), A_2' = SET(r)$.

5) Тип атрибута позволяет определить область значений этого атрибута:

$\forall a_i \in A_D, \exists D_D = DOM(a_i), D_D \subseteq D, Z(a_i) = d_D, d_D \in D_D$,

$\forall a_j \in A_I, \exists D_I = DOM(a_j), D_I \subseteq O_B, Z(a_j) = o_i, o_i \in D_I$.

6) Область определения атрибутов задается ресурсами наборов атрибутов, в которые они входят:

$\forall a_i \in A', \exists r : SET(r) = A' \Leftrightarrow RAN(a_i) = r$,

$SET(r_1) = A', SET(r_2) = A'' \Leftrightarrow RAN(a_i) = r_1 \vee r_2$.

7) Если область значений атрибута состоит из объектов, то соответствующий им информационный ресурс также является ресурсом системы:

$\forall a_j \in A_I, \exists r = TYPE(o_i), Z(a_j) = o_i$.

8) Если некоторый ресурс является подклассом информационного ресурса, то он сам является ресурсом системы:

$\forall r_2, IsRe(r_1, r_2), r_1 \in R \Rightarrow r_2 \in R$.

9) Если представление описывает некоторый ресурс, то это также представление для любого объекта соответствующего информационного ресурса: $\forall o_i \in O_B \exists P(A', v) = P(SET(r), v) = P(SET(TYPE(o_i)), v)$.

10) Если некоторому ресурсу соответствует набор атрибутов, то этот набор также соответствует любому его подресурсу: $\forall r_1, r_2 \in R, IsRe(r_1, r_2), \exists A' \subseteq S A' = SET(r_1) = SET(r_2)$.

11) Если представление описывает некоторый ресурс, то это также представление для любого его подресурса: $\forall r_1, r_2 \in R, IsRe(r_1, r_2), \exists P(A', v_i) = P(SET(r_1), v_i) = P(SET(r_2), v_i)$.

12) Если представление описывает некоторый ресурс, то это также представление для любого объекта его подресурса:

$\forall o_i \in O_B, TYPE(o_i) = r_2, IsRe(r_1, r_2), \exists P(A', v_i) = P(SET(r_1), v_i) = P(SET(r_2), v_i) = P(SET(TYPE(o_i)), v_i)$.

13) Если представление определено для некоторого набора атрибутов, то оно определено для любого информационного ресурса, соответствующего этому набору: $\forall A', A' = SET(r_1) = SET(r_2) \Rightarrow P(A', v_i) = P(SET(r_1), v_i) = P(SET(r_2), v_i)$.

14) Если некоторые ресурсы источника невозможно отобразить на ресурсы системы, то можно добавить новый информационный ресурс в систему и определить его набор атрибутов без ущерба для имеющихся ресурсов и объектов. Обратный процесс в данной постановке задачи невозможен.

15) Так как представлению объекта определенного ресурса всегда можно сопоставить определенную характеристику качества, то по этой оценке всегда можно определить качество отдельного представления объекта:

– оценка всех представлений объекта (для одного набора атрибутов) позволяет оценить «качество» объекта;

– оценка представления определенного вида для объектов некоторого типа ресурсов позволяет оценить на «соответствие определенной характеристике качества» набора объектов этого ресурса.

Примеры построения запросов

1) Извлечь все объекты определенного информационного ресурса из системы:

$\exists o_i : o_i \in O_B \wedge TYPE(o_i) = r \wedge r \in R$.

2) Извлечь все объекты определенного информационного ресурса, у которых задано значение o_i атрибута a_j и o_i является информационным объектом: $\exists o_i : o_i \in O_B \wedge TYPE(o_i) = r \wedge r \in R \wedge a_i \in A_I \wedge Z(a_j) = o_i \wedge o_i \in D_I \wedge D_I = DOM(a_j) \wedge D_I \subseteq O_B$.

3) Извлечь все объекты определенного информационного ресурса, у которых задано определенное значение o_i атрибута a_j и при этом o_i не является информационным объектом: $\exists o_i : o_i \in O_B \wedge TYPE(o_i) = r \wedge r \in R \wedge a_i \in A_D \wedge Z(a_j) = o_i \wedge o_i \in D_D \wedge D_D = DOM(a_j) \wedge \neg(D_D \subseteq O_B)$.

4) Извлечь все объекты определенного информационного ресурса из источника: $\exists o_i : TYPE(o_i) = r \wedge r \in R \wedge SET(r) = A' \wedge G^{-1}(r, A') = S_X$.

5) Извлечь все объекты определенного информационного ресурса из источника со значением o_i атрибута a_j , при этом o_i является информационным объектом: $\exists o_i : TYPE(o_i) = r \wedge r \in R \wedge SET(r) = A' \wedge G^{-1}(r, A') = S_X \wedge a_i \in A_I \wedge Z(a_j) = o_i \wedge o_i \in D_I \wedge D_I = DOM(a_j) \wedge D_I \subseteq O_B$.

6) Извлечь все объекты определенного информационного ресурса из источника со значением o_i атрибута a_j , при этом o_i не является информационным объектом: $\exists o_i : TYPE(o_i) = r \wedge r \in R \wedge SET(r) = A' \wedge G^{-1}(r, A') = S_X \wedge a_i \in A_D \wedge Z(a_j) = o_i \wedge o_i \in D_D \wedge D_D = DOM(a_j) \wedge \neg(D_D \subseteq O_B)$.

7) Запрос на представление данных о некотором объекте информационного ресурса, информация о котором может быть извлечена из разных источников: $\exists o_i : TYPE(o_i) = r \wedge r \in R \wedge SET(r) = A' \wedge G_1^{-1}(r, A') = S_{X1} \wedge \dots \wedge G_n^{-1}(r, A') = S_{Xn}$.

Рассмотрим подробнее запросы из пункта 7. Эти запросы, предназначенные для нескольких ис-

точников, возвращают $\{(o_i, S_x)\}$ – множество пар информационных ресурсов и соответствующих им источников данных. Один из вариантов представления данных пользователю предполагает просмотреть это множество в явном виде. В некоторых случаях пользователь может захотеть увидеть консолидированный объект, сформированный из результатов поиска, в таком случае решается в самом простом виде задача очистки данных: удаление дублирующих значений, объединение пересекающихся значений и т.д. Изменять структуру данных не приходится, так как эта задача решается на уровне возврата результатов с помощью функций G_i^{-1} . Как можно понять даже из такого беглого обзора, хотя ранее задача приведения и очистки явно не обозначалась, это необходимый этап при работе с источниками данных.

О задаче приведения данных

Для формального определения общей задачи приведения данных [12] вводится понятие *базиса процесса приведения* W как тройки $W = (O_B, T_W, U)$, где O_B – поток данных, T_W – множество действий, U – множество условий.

Как было сказано, O_B состоит из объектов вида $o_i = (Z(a_1), \dots, Z(a_n))$, типами значений атрибутов a_i могут быть как элементарные базовые типы, так и другие объекты из O_B . Множество $T_W = T_v \cup T_s$, где $T_v = \{t_1, \dots, t_n\}$ состоит из действий t_i , которые могут менять структуру или значения объектов $o_i \in O_B$, а множество $T_s = \{\varepsilon\}$ состоит из одного действия ε , которое никак не влияет на объекты из O_B .

Множество $U = U_{Lex} \cup U_{Syn} \cup U_{Sem}$ состоит из пересекающихся подмножеств лексических U_{Lex} , синтаксических U_{Syn} и семантических U_{Sem} – условий, накладываемых на O_B согласно структуре ресурсов, и дополнительных условий, определяемых экспертом.

Для каждого условия $u \in U$ определена пара $(\varepsilon, t) \in T_v \times T_s$, где ε определяет выполнение условия u для объектов $o_i \in O_B$, а действие $t \in T_v$ приводит те объекты из O_B , для которых условие u не выполняется, в соответствие условию u .

Для построения концептуальной модели задачи приведения данных вводятся дополнительные сущности для оценки эффективности приведения и ее оптимизации, определяются связи между введенными сущностями, вводятся *ограничения* на допустимые сочетания компонентов модели и некоторые их свойства. Определяются условия для выделения *допустимого порядка* применения действий для множества рассматриваемых условий, при котором достигается лучший результат. Эффективное решение задач приведения прямым образом оказывает влияние на качество данных и их оценку. В свою очередь, оценка качества данных позволяет определить, насколько эффективно будут выпол-

няться задачи, решаемые в рамках семантической библиотеки, например, задачи поиска идентичных ресурсов в источниках данных для связывания данных.

При этом сохранение для пользователя данных об объектах информационного ресурса из разных источников может быть выполнено двумя способами:

1) **связывание** – этот способ идентичен по смыслу проставлению связи «смотри также» и означает, что на одном конце содержится более полная и обширная информация по ресурсу, а на другом – либо уже находящийся в системе информационный объект, либо вновь создаваемый; при этом сохраняются как минимум значения идентифицирующих свойств;

2) **идентификация** означает, что на одном конце содержится точно такой же по качеству информации объект, как и на другом; при этом информационный объект может уже находиться в системе до выявления идентичного в источнике данных либо просто создается новый объект, который фактически представляет собой ссылку на объект источника.

Качество данных

Качество данных является важной характеристикой любого набора данных независимо от его структуры. Опираясь на оценку качества данных, можно давать объективную оценку эффективности процессов, происходящих в семантической библиотеке, наиважнейшим из которых является интеграция данных с другими источниками, необязательно являющимися библиотеками.

Исходя из определения атрибутов и их видов при решении, например, задач связывания данных из разных источников, необходимо предварительно оценить как минимум качество значений идентифицирующих атрибутов. В противном случае сложно объективно проанализировать полученные результаты. Поэтому используем формальное определение качества, удовлетворяющее в контексте данной задачи.

Если данные пригодны для предназначенных для них целей, то можно сказать, что эти данные качественные [13]. Качество данных определяется связанными характеристиками, каждая из которых определяет наличие тех или иных недостатков во множестве данных [14].

Если O_B – множество объектов, используем обозначения: $o_1, o_2 \in O_B$, где $TYPE(o_1) = r$, $TYPE(o_2) = r$, a_i – атрибут из набора $SET(r) = A.val(Z(a_{1i}), Z(a_{2i})) = true$, если значения атрибутов a_{1i} и a_{2i} объектов o_1 и o_2 совпадают, $f(a_j) = true$, $i \leq n$, если задан формат значений a_j , которые могут быть представлены в виде последовательности символов из непустого конечного алфавита L , то есть для таких значений a_j существует грамматика G_j , которая задает значе-

ния атрибута a_j , $U = \{u_1, \dots, u_n\}$ – множество условий, накладываемых на объекты, $u(a_i, \dots, a_k)$ будем обозначать как $g(o_i, u_s)$ – функцию, принимающую значения $true$ в случае выполнения условия u_s для объекта и $false$ в противном случае.

Будем говорить о наличии на уровне данных [13, 14]:

а) *синтаксических ошибок* (обозначим через M_{Syn}), если для

$Z(a_i) = null$, при этом не существует условия вида $u(a_i, a_j)$, когда от a_i нет зависимых значений, $Z(a_i) \notin RAN(a_i)$, $f(a_j) \neq true$, $a_j \in A_D$ и $a_j \notin L(G_J)$.

б) *семантических ошибок* (обозначим через M_{Sem}), если выполняется

$Z(a_i) = null$, при этом существует условие вида $u(a_i, a_j)$, такое что $Z(a_j) \neq null$,

$g(a_i, u_j) \neq true$, $a_j \in A$,

если для любого o_1 существует $o_2 \in O_B$ и $val(o_1, o_2) = \wedge val(Z(a_{1i}), Z(a_{2i})) = true$;

в) *ошибок на уровне схем данных* [1, 2, 4, 7], если $G(r_X, A_r) \neq (r, A')$, то есть не существует функции отображения ресурсов источника на ресурсы системы.

Определим основные характеристики качества уровня данных.

Точность (Acc). Точные данные не содержат никаких ошибок, кроме, может быть, дублирования и недостающих значений.

Синтаксическая точность (или *согласованность* – Acc_{Syn}).

Если $\forall a_j \in A_D$ и $a_j \in L(G_J)$ выполняется $f(a_j) = true$, то o_i – синтаксически точный объект или $Acc_{Syn}(o_i) = true$.

Если $\forall o_k \in O_B$ выполняется $Acc_{Syn}(o_k) = true$, то есть все объекты синтаксически точны, то множество O_B называется *синтаксически точным множеством данных*: $Acc_{Syn}(O_B) = \wedge Acc_{Syn}(o_k) = true$.

Семантическая точность (или *достоверность* Acc_{Sem}).

Если $\forall u_j$ выполняется $g(o_k, u_j) = true$, то o_k – семантически точный объект или $Acc_{Sem}(o_i) = true$.

Если $\forall o_k \in O_B$ выполняется $Acc_{Sem}(o_k) = true$, то есть все объекты семантически точны, то множество O_B называется *семантически точным множеством данных*: $Acc_{Sem}(O_B) = \wedge Acc_{Sem}(o_k) = true$.

Если o_k – точный объект, то $Acc(o_k) = Acc_{Syn}(o_k) \wedge Acc_{Sem}(o_k) = true$.

Если O_B – точное множество, то $Acc(O_B) = Acc_{Syn}(O_B) \wedge Acc_{Sem}(O_B) = true$.

Полнота (F). Полные данные не содержат недостающих значений и объектов.

Синтаксическая полнота (F_{Syn}).

Если $\forall o_k \in O_B$ выполняется $Z(a_i) \in RAN(a_i)$ и $Z(a_i) \neq null$, при этом не существует условия вида $u(a_i, a_j)$, то o_k – синтаксический полный объект или $F_{Syn}(o_k) = true$.

Если $\forall o_k \in O_B$ выполняется $F_{Syn}(o_k) = true$, то есть все объекты синтаксически полные, то множество

O_B называется синтаксически полным множеством данных: $F_{Syn}(O_B) = \wedge F_{Syn}(o_k) = true$.

Семантическая полнота (F_{Sem}).

Если $\forall o_k \in O_B$ выполняется $Z(a_i) \in RAN(a_i)$ и $Z(a_i) \neq null$, при этом существует условие вида $u(a_i, a_j)$, то o_k – семантически полный объект или $F_{Sem}(o_k) = true$.

Если $\forall o_k \in O_B$ выполняется $F_{Sem}(o_k) = true$, то есть все объекты семантически полные, то множество O_B называется семантически полным множеством данных:

если o_k – полный объект, то $F(o_k) = F_{Syn}(o_k) \wedge F_{Sem}(o_k) = true$,

если O_B – полное множество, то $F(O_B) = F_{Syn}(O_B) \wedge F_{Sem}(O_B) = true$.

Уникальность (Uniq). Если $\forall o_1$ не существует $o_2 \in O_B$ и $val(o_1, o_2) = \wedge val(Z(a_{1i}), Z(a_{2i})) = true$, то o_1 называется уникальным объектом или $Uniq(o_1) = true$. Если все объекты уникальны, то множество O_B называется *уникальным множеством данных*.

Таким образом, определены ошибки только двух типов. Далее будем использовать обозначения: M – ошибки, M_{Syn} – синтаксические ошибки и M_{Sem} – семантические ошибки, такие что $M_{Sem} \subseteq M$, $M_{Syn} \subseteq M$, $M_{Syn} \subseteq M_{Sem}$, каждый тип ошибок разбивается на подтипы в зависимости от того, какому критерию качества они соответствуют.

Зависимость уровня качества данных от порядка обработки ошибок

Уровень качества данных определяется степенью их соответствия характеристикам, используемым при определении понятия «качество данных». Наилучшее качество достигается при полном соответствии всем характеристикам.

Для достижения высокого уровня качества данных прежде следует обрабатывать синтаксические ошибки, а затем семантические. При определении синтаксических ошибок при их обработке происходит анализ отдельных значений атрибутов объектов независимо от других значений, объектов и их связей, тогда как из определения семантических ошибок при их обработке происходит анализ значений атрибутов, зависимых также от других условий. Поэтому синтаксическая корректность значений влияет на качество обработки семантических ошибок, так как часть ошибок могут оказаться «невидимыми».

Последовательность обработки ошибок разных подтипов во множествах M_{Syn} , M_{Sem} определяется в зависимости от уровня вложенности соответствующей характеристики качества и приоритетов, присвоенных характеристикам одного уровня (чем меньше значение приоритета, тем позже обрабатывается соответствующая ошибка). Причем, если некоторым характеристикам одного уровня присвоен одинаковый приоритет, предпочтение отдается той характеристике, приоритет предка которой выше.

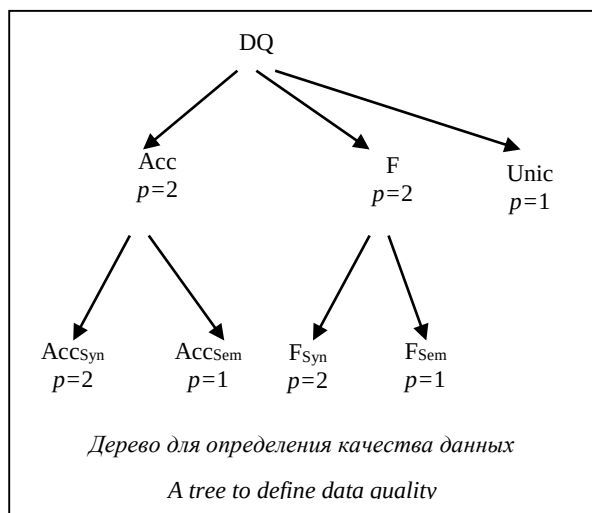
Если же приоритеты предков совпадают, определение порядка обработки не может существенно повлиять на результат (то есть порядок обработки ошибок в каждом подобном случае может меняться), так как значения, используемые при обработке соответствующих ошибок, не могут зависеть от значений друг друга по определению.

Определение порядка обработки ошибок происходит в несколько шагов:

- $order(M) = (M_{Syn} \rightarrow M_{Sem})$,
- $order(M_{Syn}) = ((p_1 \rightarrow \dots \rightarrow p_k)_1, \dots, (p_1 \rightarrow \dots \rightarrow p_h)_m)$,
- $order(M_{Sem}) = ((p_1 \rightarrow \dots \rightarrow p_l)_1, \dots, (p_1 \rightarrow \dots \rightarrow p_s)_m)$,

где p_i – приоритет характеристик качества данных, соответствующих обрабатываемым ошибкам, $1 \leq i \leq k$, $1 \leq i \leq h$, $1 \leq i \leq l$, $1 \leq i \leq s$, где k, h, l, s – количество используемых характеристик качества данных одного уровня; $(p_1 \rightarrow \dots \rightarrow p_i)_i$ – порядок обработки ошибок, соответствующих характеристикам, расположенным на одном уровне согласно приоритету, где $1 \leq i \leq m$, m – наибольший уровень вложенности используемых характеристик в дереве, корень дерева имеет уровень 0. Если обрабатывается несколько ошибок, соответствующих одной характеристике в одном подмножестве потока данных, то порядок их обработки в рамках этой характеристики определяется произвольно, так как зависимости между значениями, участвующими в анализе этих ошибок, не влияют на результат по определению.

Для используемого определения качества данных построим соответствующее ему дерево (см. рисунок).



В заключение отметим, что в работе даны формальные определения для описания контента семантической библиотеки, являющегося ядром любой библиотеки. Авторы исходили из того, что структура данных библиотеки будет определяться не на этапе проектирования библиотеки, а в процессе ее эксплуатации. При этом важной состав-

ляющей библиотеки является возможность интегрирования данных с разными источниками, соответственно, процесс интеграции данных является одним из ключевых. На этапе интеграции основную роль играет функция G , которая отвечает за преобразование данных источника. Фактически эта функция разбивает процесс интеграции в соответствии с определением характеристик качества на более мелкие подзадачи, соответствующие задачам приведения данных. Обычно слабое место в задачах приведения – это выполнение, которое должно быть эффективным даже на больших объемах данных, так как операции приведения данных имеют достаточно высокую вычислительную сложность и высокие временные затраты. Поэтому для оценки качества данных в конкретном случае могут использоваться не все характеристики, а только некоторые, для того чтобы достичь приемлемого качества данных за приемлемое время. Соответственно функция G описывает процесс интеграции в каждом отдельном случае по-разному.

В следующей работе планируется описать онтологию, основанную на этих понятиях, и подробнее остановиться на задачах описания, интеграции и обработки данных в терминах этой онтологии. Возможность описания понятийной модели ресурсов библиотеки в терминах этой онтологии в совокупности с поддержкой тезауруса этой предметной области является, на взгляд авторов, необходимым условием для создания гибких семантических библиотек, интегрируемых в LOD и использующих все возможные преимущества взаимосвязанных данных из его источников.

Литература

1. Антопольский А.Б., Каленкова А.А., Каленов Н.Е., Серебряков В.А., Сотников А.Н. Принципы разработки интегрированной системы для научных библиотек, архивов и музеев // Информационные ресурсы России. 2012. № 1. С. 2–7.
2. Горный Е., Вигурский К. Развитие электронных библиотек: мировой и российский опыт, проблемы, перспективы // Интернет и российское общество; [под ред. И. Семенова]. М.: Гендальф, 2002. С. 158–188.
3. Серебряков В.А., Атаева О.М. Персональная цифровая библиотека Libmeta как среда интеграции связанных открытых данных // RCDL. 2014.
4. Bizer C., Heath T., and Berners-Lee T. Linked data – the story so far. Int. J. Semantic Web Inf. Syst., 2009, no. 5 (3), pp. 1–22.
5. Semantic Web. URL: <http://www.w3.org/standards/semanticweb/> (дата обращения: 10.06.2015).
6. Resource Description Framework URL: <http://www.w3.org/RDF/> (дата обращения: 10.06.2015).
7. Gruber T.R. A translation approach to portable ontologies. Knowledge Acquisition, 1993, no. 5 (2), pp. 199–220.
8. Lihua Zhao. Ontology Integration for the Linked Open Data. 2013 URL: <http://ri-www.nii.ac.jp/~lihua/PhD.Thesis.Lihua-Zhao.pdf> (дата обращения: 10.06.2015).
9. Электронная библиотека «Научное наследие России». URL: <http://e-heritage.ru/index.html> (дата обращения: 10.06.2015).
10. Sebastian Ryszard Kruk • Bill McDaniel. Semantic Digital Libraries. Springer-Verlag Berlin Heidelberg, 2009.
11. Candela L., Castelli D., Dobrev M., Ferro N., Ioannidis Y.,

Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Schuldts H., Soergel D. The DELOS Digital Library Reference Model Foundations for Digital Libraries. IST-2002 2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. Version 0.98, December 2007. URL: http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf (дата обращения: 10.06.2015).

12. Rahm E., Hai Do H. *Data Cleaning: Problems and Current Approaches*. 2000.

13. Scannapieco M., Missier P., Batini C. Data Quality at a Glance. *Datenbank-Spektrum*, 2005, vol. 14, pp. 6–14.

14. Singh R., Dr. Singh K. A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing. *IJCSI Intern. Journ. of Computer Sc. Issues*, 2010, vol. 7, iss. 3, no. 2.

DOI: 10.15827/0236-235X.112.180-187

Received 15.06.15

THE BASIC CONCEPTS OF A SEMANTIC LIBRARIES FORMAL MODEL AND ITS INTEGRATION PROCESS FORMALIZATION

Ataeva O.M., Junior Researcher, oli@ultimeta.ru;

Serebryakov V.A., Dr.Sc. (Physics and Mathematics), Professor, Head of Department, serebr@ultimeta.ru
(A.A. Dorodnitsyn Computing Centre of RAS, Vavilov St. 40, Moscow, 119333, Russian Federation)

Abstract. Modern technology development leads to redefinition of the library content concept, which may be not only a traditional description of printed publications, but also other types of objects. The content of digital libraries and physical objects can be linked in various ways. This article considers the library as a structured data repository suitable for integration with other data sources. The paper presents a thesaurus structure to determine their thematic scope, as well as basic concepts to describe such libraries. Such concepts as information resources, attributes sets, information and other associated objects form a conceptual basis for a generated semantic library subject domain. Thesaurus, in its turn, provides terminology support for these concepts, facilitates information system navigation, supports the process of user query qualification and completion. The article also describes the concepts for detailed data integration work; the emphasis is on the concepts used in the process of data reduction. An important characteristic of any data set regardless of its structure is data quality. Based on data quality assessment it is possible to give an objective assessment of the semantic library process efficiency, where the most important process is data integration with other sources. A formal model of concepts described in the article is used to describe a library ontology.

Keywords: semantic library, data quality, data reduction, data model.

References

1. Antopolsky A.B., Kalenkova A.A., Kalenov N.E., Serebryakov V.A., Sotnikov A.N. Integrated system developing principles for scientific libraries, archives and museums. *Informatsionnye resursy Rossii* [Russian Information Resources]. 2012, no. 1, pp. 2–7 (in Russ.).
2. Gorny E., Vigursky K. E-libraries development: world and Russian experience, problems, prospects. *Internet i rossyskoe obshchestvo* [The Internet and Russian Society]. I. Semenov (Ed.). Moscow, Gendalf Publ., 2002, pp. 158–188 (in Russ.).
3. Serebryakov V.A., Ataeva O.M. Libmeta personal digital library as an integrated environment of connected open data. *Proc. of RCDL-2014*. Available at: <http://rcdl2014.jinr.ru/> (accessed June 10, 2015).
4. Bizer C., Heath T., Berners-Lee T. Linked data – the story so far. *Int. J. Semantic Web Inf. Syst.* 2009, no. 5 (3), pp. 1–22.
5. *Semantic Web*. Available at: <http://www.w3.org/standards/semanticweb/> (accessed June 10, 2015).
6. *Resource Description Framework (RDF)*. Available at: <http://www.w3.org/RDF/> (accessed June 10, 2015).
7. Gruber T.R. A translation approach to portable ontologies. *Knowledge Acquisition*. 1993, no. 5 (2), pp. 199–220.
8. Zhao L. *Ontology Integration for the Linked Open Data*. 2013. Available at: <http://ri-www.nii.ac.jp/~lihua/PhD.Thesis.LihuaZhao.pdf> (accessed June 10, 2015).
9. *Elektronnaya biblioteka "Nauchnoe nasledie Rossii"* [E-library "Scientific Heritage of Russia"]. Available at: <http://e-heritage.ru/index.html> (accessed June 10, 2015).
10. Kruk S.R., McDaniel B. *Semantic Digital Libraries*. Springer-Verlag Berlin Heidelberg Publ., 2009.
11. Candela L., Castelli D., Dobрева M., Ferro N., Ioannidis Y., Katifori H., Koutrika G., Meghini C., Pagano P., Ross S., Agosti M., Schuldts H., Soergel D. The DELOS Digital Library Reference Model Foundations for Digital Libraries. IST-2002 2.3.1.12. Technology-enhanced Learning and Access to Cultural Heritage. Version 0.98. 2007. Available at: http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf (accessed June 10, 2015).
12. Rahm E., Hai Do H. *Data Cleaning: Problems and Current Approaches*. 2000.
13. Scannapieco M., Missier P., Batini C. Data Quality at a Glance. *Datenbank-Spektrum*. 2005, vol. 14, pp. 6–14.
14. Singh R., Dr. Singh K. A Descriptive Classification of Causes of Data Quality Problems in Data Warehousing. *IJCSI Int. Journ. of Computer Science Issues*. 2010, vol. 7, iss. 3, no. 2.

УДК 519.68; 681.513.7; 612.8.001.57; 007.51/.52
DOI: 10.15827/0236-235X.112.188-197

Дата подачи статьи: 02.07.15

ПРОГРАММНЫЙ КОМПЛЕКС ДЛЯ АНАЛИЗА ДАННЫХ ИЗ СОЦИАЛЬНЫХ СЕТЕЙ

Т.В. Батура, к.ф.-м.н., старший научный сотрудник, tatiana.v.batura@gmail.com;

Ф.А. Мурзин, к.ф.-м.н., зам. директора по науке, murzin@iis.nsk.su;

*А.В. Проскуряков, аспирант, alexey.proskuryakov@gmail.com
(Институт систем информатики им. А.П. Ершова СО РАН,
просп. Лаврентьева, 6, г. Новосибирск, 630090, Россия)*

Статья посвящена проблемам извлечения и обработки данных из социальных сетей. Рассмотрены различные количественные характеристики, отношения и множества, вычисляемые на основе получаемых данных. Важно, что эти характеристики являются конструктивными и могут быть эффективно вычислены или построены при помощи соответствующих алгоритмов.

Количество информации, находящейся в социальных сетях, очень велико. При использовании распределенной системы извлечения и обработки данных объем увеличивается еще больше. Поэтому наиболее важной и трудной задачей является выделение той части данных, которую можно было бы достаточно эффективно обработать и которая представляла бы интерес в соответствии с поставленными целями. Для решения этой задачи предлагается применить метод определения количества влияния аудитории на отдельного пользователя. Метод основан на теории динамического социального влияния, предложенной Б. Латане. Данный подход является полезным также при решении задачи определения источника распространения информации.

Задача определения количества влияния на пользователя со стороны других пользователей непосредственно связана с задачей обнаружения лидеров мнений – сравнительно популярных пользователей, которые формируют мнение остального большинства. В работе предложен один из возможных методов определения лидеров мнений.

Приведено довольно подробное описание разработанного программного комплекса, позволяющего работать с информацией из социальных сетей ВКонтакте и Twitter. Он состоит из шести модулей: извлечения, обработки данных, отслеживания изменений пользовательских данных, анализа данных, построения графовых структур и модуля визуализации данных.

Ключевые слова: анализ социальных сетей, обработка данных на естественном языке, источник распространения информации, лидеры мнений, межличностный анализ.

Сложно представить современный мир без социальных сетей. Это удобное средство общения между людьми, кроме того, сети открывают новые возможности для анализа потоков информации и поведения людей в процессе общения. Анализ структуры графов и текстовых данных, получаемых из социальных сетей, является наиболее эффективным методом изучения взаимодействий между их участниками. Важный элемент подобных исследований – создание программных инструментов, позволяющих извлекать из сетей данные и обрабатывать их.

В статье приведено довольно подробное описание разработанного программного комплекса для извлечения, обработки и анализа пользовательских данных. Эксперименты проводились по извлечению данных из сетей ВКонтакте и Twitter, хотя модуль извлечения системы имеет возможность функционального расширения практически на любую социальную сеть в зависимости от предоставляемого API. В целом система состоит из шести модулей, в том числе модуля отслеживания изменений в пользовательских данных и модуля построения графов.

Просматривать и обрабатывать невероятно большие потоки информации, которые постоянно обновляются, не только затруднительно, но и нерационально. По мнению авторов, сравнительно

большая часть информации в социальных сетях представляет собой средство самовыражения, когда человек делает заметки для себя, не преследуя какие-то дополнительные цели наподобие рекламы, агитации и пр. Это предположение подтверждает официальная статистика [1], согласно которой, 40 % всех твитов можно отнести к светской беседе.

Чтобы выбрать существенную информацию из большого объема разнородных данных, необходимо разрабатывать определенные критерии. В ряде случаев достаточно ограничить число отслеживаемых пользователей и, к примеру, просматривать только тех, которые пользуются большей популярностью. Их называют лидерами мнений. Это люди, чье мнение оказывает на окружающую аудиторию большое влияние, формируя таким образом мнение остального сообщества.

Помимо количества подписчиков или друзей, лидера мнений характеризуют такие величины, как количество созданных сообщений, лайков, репостов и комментариев, которые собирают записи пользователя.

Не менее важной задачей является определение количества влияния, оказываемого на человека со стороны его окружения. Для ее решения предлагается адаптировать к социальным сетям теорию Латане социального влияния [2, 3].

Количественные характеристики, отношения и множества, вычисляемые на основе данных из социальных сетей

При анализе социальных сетей целесообразно рассматривать ряд числовых и нечисловых характеристик, отношений и множеств, естественно связанных с пользователями сети и сообщениями, циркулирующими в ней. Важно, чтобы эти характеристики могли быть вычислены или построены при помощи соответствующих алгоритмов.

Обозначим T сообщение (твит) социальной сети, u – пользователя сети, который может создавать и пересылать сообщения.

Одноместные характеристики:

$Followers_Count(u)$ – количество людей, которые читают сообщения данного пользователя (то есть подписчиков этого пользователя);

$Friends_Count(u)$ – количество друзей данного пользователя (пользователь сам заносит некоторых людей в список друзей);

$Retweets(T)$ – количество пересылок данного сообщения.

Множества:

$Followers(u)$ – подписчики данного пользователя;

$Friends(u)$ – друзья данного пользователя;

$Mentions(u)$ – имена пользователей, упоминаемые в сообщениях данного пользователя;

$Hashtags(u)$ – хэштеги, встречающиеся в сообщениях данного пользователя;

$Urls(u)$ – внешние ссылки, встречающиеся в сообщениях данного пользователя.

Хэштег – это слово или набор слов, записанных без пробелов и начинающихся с символа $\#$. Является одной из форм метаданных. Короткие сообщения в блогах или социальных сетях типа Facebook, Twitter или Instagram могут быть помечены символом $\#$ перед ключевыми словами или фразами, не содержащими пробелов. Они встречаются в любых предложениях. Например, «Эверест (Гималаи) самая высокая точка мира $\#12$ МестКоторыеЯБыХотелПосетить». Хэштеги позволяют группировать похожие сообщения. По заданному хэштегу можно найти набор сообщений, содержащих его.

Числовые характеристики, ассоциированные с множествами:

$Count_Mentions_u(v)$ – количество упоминаний пользователя v пользователем u ;

$Count_Hashtags_u(v)$ – количество употреблений хэштега v пользователем u ;

$Count_Urls_u(v)$ – количество упоминаний внешней ссылки v пользователем u ;

$Count_Retweets_u(u_1)$ – количество сообщений, пересланных пользователем u , полученных от пользователя u_1 .

При анализе процессов, происходящих в коллективах, в исследованиях по социологии, психологии, экономике часто рассматривают трехмест-

ные отношения предпочтения [4]. Согласно [5], запись $i \vdash_k j$ означает, что i предпочтительнее j , по мнению k . Критерии предпочтительности могут быть самыми разными: профессионализм, который можно разбить на разные виды деятельности, что породит спектр новых критериев; умение руководить людьми; коммуникабельность; восприимчивость к инновациям; психологическая устойчивость и т.д. На базе этих отношений складывается неформальная структура коллектива, важность выявления которой в ряде случаев не требует комментариев.

Трехместные отношения:

$Mentions_u(u_1, u_2)$ – пользователь u упоминает пользователя u_1 не реже, чем u_2 ;

$Hashtags_u(h_1, h_2)$ – пользователь u употребляет хэштег h_1 не реже, чем хэштег h_2 ;

$Urls_u(url_1, url_2)$ – пользователь u упоминает ссылку url_1 не реже, чем ссылку url_2 ;

$Retweets_u(u_1, u_2)$ – пользователь u пересылает сообщения, полученные от пользователя u_1 , не меньше число раз, чем полученные от пользователя u_2 .

Числовые характеристики, ассоциированные с трехместными отношениями:

$N_Mentions_u(u_1, u_2) =$

$= Count_Mentions_u(u_1) - Count_Mentions_u(u_2);$

$N_Hashtags_u(h_1, h_2) =$

$= Count_Hashtags_u(h_1) - Count_Hashtags_u(h_2);$

$N_Urls_u(url_1, url_2) =$

$= Count_Urls_u(url_1) - Count_Urls_u(url_2);$

$N_Retweets_u(u_1, u_2) =$

$= Count_Retweets_u(u_1) - Count_Retweets_u(u_2).$

Можно считать, что приведенные выше функции также позволяют определять силы влияния различных факторов. Например, функция $N_Mentions_u(u_1, u_2)$ позволяет вычислить силу влияния пользователя u_1 по сравнению с u_2 на пользователя u , то есть силу влияния с учетом предпочтений пользователя u . Например, если пользователь $u = "@mursya_ru"$ упоминает 8 раз пользователя $u_1 = "@tadatuda"$ и 12 раз пользователя $u_2 = "@voishev"$, то $N_Mentions_u(u_1, u_2) = 12 - 8 = 4$.

Теория Латане социального влияния и ее модификация

Теория динамического социального влияния, предложенная Латане [2, 3], позволяет вычислить «количество влияния», оказанного на получателя информации из нескольких источников. Латане выделял три наиболее важных атрибута отношений между получателем информации и ее источниками: сила источника влияния на агента, расстояние между агентом и источником и количество источников, влияющих на получателя. Под расстоянием понимается оперативность перемещения информации от источника к агенту воздействия.

Согласно теории социального влияния, уровень влияния, испытываемого агентом, может быть

выражен следующей формулой: $I_i = -S_i\beta - \sum_{j=1, j \neq i}^N \frac{S_j O_j O_i}{d_{i,j}^\alpha}$, где I_i – количество социального давления, направленного на агента i ; O_i – мнение i -го агента (± 1) по отношению к данному вопросу, значение $+1$ соответствует поддержке, а -1 – сопротивлению предложению; S_i – сила социального влияния ($S_i \geq 0$); β – сопротивляемость агента к изменениям ($\beta > 0$); d_{ij} – расстояние между агентами i и j ($d_{ij} \geq 1$); α – степень ослабления расстояния ($\alpha \geq 0$); N – общее число взаимодействующих агентов.

Значение постоянной β обычно принимается равным 2 в соответствии с величиной, использованной в исследованиях Латане. Большее значение этой постоянной означает, что для изменения мнения требуется большее давление, меньшее значение соответствует меньшему усилию. Значение постоянной α также обычно принимается равным 2. Большие значения α означают, что с ростом расстояния между источником и получателем требуется гораздо большая величина давления.

Величина d_{ij} определяется свойствами пары агентов, она может рассматриваться как показатель легкости общения (передачи информации). При задании данной величины могут учитываться возрастные, национальные, конфессиональные и другие различия. Формула для вычисления d_{ij} может включать в себя физическое расстояние. Например, между населенными пунктами, в которых находятся агенты. Обычно учитывается то, что легкость коммуникации подчиняется закону об обратной квадратичной зависимости от физического расстояния [3]. В случае с социальными сетями возможны различные подходы, в том числе такие, когда физическое расстояние не принимается во внимание.

Для анализа социальных сетей авторы предлагают модификацию формулы Латане в следующем виде:

$$I_u = -\beta \cdot \sum_{i=1}^N \text{Count_Mentions}_u(u_i) - \sum_{i=1}^N \sum_{j=2, j > i}^N \frac{N_Mentions_u(u_i, u_j)}{\rho^\alpha(u_i, u_j)}.$$

В этой формуле учитываются все пользователи, упоминаемые u . Можно считать, что все они на него влияют. В следующей формуле учитывается влияние только наиболее упоминаемого и наименее упоминаемого пользователей:

$$I_u^1 = -\beta \cdot \max_{i=1}^N \{ \text{Count_Mentions}_u(u_i) \} - \max_{i=1}^N \max_{j=2, j > i}^N \left\{ \frac{N_Mentions_u(u_i, u_j)}{\rho^\alpha(u_i, u_j)} \right\}.$$

Также можно учитывать влияние только наиболее упоминаемого и следующего за ним по частоте упоминания:

$$I_u^2 = -\beta \cdot \max_{i=1}^N \{ \text{Count_Mentions}_u(u_i) \} - \min_{i=1}^N \min_{j=2, j > i}^N \left\{ \frac{N_Mentions_u(u_i, u_j)}{\rho^\alpha(u_i, u_j)} \right\}.$$

Здесь $\rho(u_i, u_j)$ – расстояние между пользователями u_i и u_j . В формуле учитываются все пользователи, упоминаемые u , то есть считается, что все они на него влияют. Расстояние можно, например, задать отношением «подписчик–подписчик подписчика–подписчик подписчика подписчика» и т.д. Так как эти выкладки производятся относительно предпочтений пользователя u , уместно будет воспользоваться французской железнодорожной метрикой:

$$\rho(u_i, u_j) = \begin{cases} \|u_i - u_j\|, & u_i - u = \lambda(u_j - u), \\ \|u_i - u\| + \|u_j - u\|, & u_i - u \neq \lambda(u_j - u), \end{cases}$$

где λ – заданный коэффициент; u – фиксированная выбранная точка, через которую обязательно должен проходить путь между u_i и u_j . Самое простое – подсчитать количество ребер. Можно, конечно, приписывать вес каждому ребру. Но, на взгляд авторов, здесь вес ребра не столь важен, так как в текущий момент времени пользователь u может не быть подписчиком, например, пользователя u_j и поэтому находиться от него на далеком расстоянии, а в следующий момент времени уже стать подписчиком.

Внешнее влияние, например влияние СМИ, также может быть учтено, если в основную формулу Латане добавить дополнительное слагаемое $-O_i O_M S_{Mi}$, где S_{Mi} – сила влияния внешних источников на агента i ($S_{Mi} > 0$); O_M – мнение внешнего источника (± 1). Для учета влияния масс-медиа Латане [2, 3] получил итоговую формулу

$$I_i = -S_i\beta - O_i O_M S_{Mi} - \sum_{j=1, j \neq i}^N \frac{S_j O_j O_i}{d_{i,j}^\alpha}.$$

Обычно внешний источник также моделируется как агент, но «вне окружающей среды» и с расстоянием 1 до каждого агента ввиду своей «вездесущей» природы. Величина S_{Mi} меняется в зависимости от агента, так как каждый человек испытывает различное давление СМИ. Эта величина аналогична иногда рассматриваемой «величине доверия» агента к сообщениям, получаемым из внешних источников. Для социальных сетей аналогом СМИ можно считать хэштеги и внешние ссылки. Соответственно получаем формулу

$$I_u = -\beta \cdot \sum_{i=1}^N \text{Count_Mentions}_u(u_i) - \sum_{i=1}^N \sum_{j=2, j > i}^N \frac{|\text{Hashtags}(u_i)| |\text{Hashtags}(u_j)|}{\rho^\alpha(u_i, u_j)} - \sum_{i=1}^N \sum_{j=2, j > i}^N \frac{N_Mentions_u(u_i, u_j)}{\rho^\alpha(u_i, u_j)},$$

в которой учитываются все пользователи, упоминаемые u , и все хэштеги.

Аналогично получаем формулу, в которой учитываются все пользователи и внешние ссылки в Интернете, упоминаемые пользователем u :

$$I_u = -\beta \cdot \sum_{i=1}^N \text{Count_Mentions}_u(u_i) - \\ - \sum_{i=1}^{|Urls(u)|} \sum_{\substack{j=2 \\ i > j}}^{|Urls(u)|} \text{Urls}_u(url_i, url_j) - \\ - \sum_{i=1}^N \sum_{\substack{j=2 \\ i > j}}^N \frac{N_Mentions_u(u_i, u_j)}{\rho^a(u_i, u_j)}.$$

Обнаружение лидеров мнений

По данным сайта Statista [6], социальная сеть ВКонтакте насчитывает 55 млн активных российских пользователей, Twitter – 8,5 млн человек [7]. Среди жителей Казахстана на долю ВКонтакте приходится 3,8 млн пользователей, Twitter используют 300 тысяч человек [8]. Количество пользователей социальных сетей постоянно увеличивается. Для сравнения: Twitter сейчас насчитывает 236 млн пользователей во всем мире [9].

Просматривать и обрабатывать огромные потоки информации, которые постоянно обновляются, не только трудно, но и нерационально. По мнению авторов, сравнительно большая часть информации в социальных сетях представляет собой средство самовыражения, когда человек делает заметки для себя, не преследуя какие-то дополнительные цели типа рекламы, агитации и пр. Это предположение подтверждает официальная статистика [1], согласно которой 40 % всех твитов можно отнести к светской беседе. Вместе с тем следует особо рассматривать пользователей Twitter, которые являются официальными представителями группы людей в сети. Например, новостные каналы (радио, телевидение, пресса), рекламные агентства, различные организации, в том числе государственного уровня, и так далее. В отличие от Twitter в сети ВКонтакте существует в явном виде понятие сообщества, то есть группы людей, объединенных интересами. Безусловно, некоторая часть аккаунтов, зарегистрированных как один самостоятельный пользователь ВКонтакте, в реальной жизни может управляться группой людей.

Чтобы выбрать существенную информацию из большого объема разнородных данных, необходимы критерии, позволяющие анализировать самое важное. Таким образом можно ограничить число просматриваемых пользователей. Другими словами, хотелось бы просматривать только сравнительно популярных пользователей, которые формируют мнение остального большинства. Их называют лидерами мнений. Показателем популярности здесь может выступать количество подпис-

чиков пользователя. Согласно новому закону о блоггерах (№ 97-ФЗ от 05.09.14), автор с аудиторией свыше 3 000 пользователей в сутки должен заноситься в спецреестр.

Но большое количество подписчиков не является достаточно точным показателем того, что пользователь – лидер мнений. Количество публикуемых сообщений тоже важно, причем созданных самим пользователем, а не просто перепостов. Кроме того, необходимо учитывать реакцию аудитории в виде количества комментариев к собранному сообщению, количества лайков и ретвитов. Таким образом, пользователя Twitter, которого можно назвать лидером мнений, характеризует следующий набор величин в терминах извлекаемых нами данных:

- количество подписчиков (followers_count);
- количество самостоятельно созданных твитов (statuses_count);
- количество репостов сообщения пользователя (retweet_count);
- количество лайков, которые собрал твит пользователя (favorite_count).

Вместо количества самостоятельно созданных твитов удобнее оперировать величиной, вычисляемой как отношение самостоятельно созданных твитов к общему количеству сообщений пользователя,

то есть $\frac{ST}{TT}$, где ST – количество самостоятельно созданных твитов; TT – общее количество сообщений пользователя.

Заметим, что перед текстом твит-сообщения может присутствовать пометка RT , означающая, что сообщение создано не лично пользователем, а является перепостом уже существующего. Это позволяет точно вычислить величину ST .

Количество репостов сообщения (retweet_count) и количество лайков, которые собрал твит (favorite_count), являются характеристиками самого сообщения, а не его автора, поэтому не могут вносить непосредственный вклад в оценку популярности пользователя. Чтобы их учесть, можно, например, применить к каждой из этих характеристик алгоритм, аналогичный вычислению индекса Хирша [10]. В итоге количеству репостов сообщения пользователя будет соответствовать коэффициент χ_1 , а количеству лайков, которые собрал твит пользователя, – коэффициент χ_2 . Эти два коэффициента уже могут использоваться в качестве критерия популярности пользователя.

Лидера мнений в сети ВКонтакте характеризует следующий набор величин в терминах извлекаемых данных:

- количество подписчиков (getProfiles().counters.followers);
- количество друзей (getProfiles().counters.friends);

- количество записей пользователя (`getProfiles().counters.notes`);
- количество лайков, собранных записью пользователя (`likes_count`);
- количество комментариев, собранных записью пользователя (`comments_count`).

Следует отметить, что количество друзей в сети ВКонтакте – более важный признак популярности пользователя, чем количество подписчиков. В сети Twitter ситуация противоположная.

В сети ВКонтакте дополнительной характеристикой сообщения является количество комментариев, которые собрала запись пользователя. К ней также можно применить алгоритм, аналогичный вычислению индекса Хирша [10]. В результате получим величину χ_3 , которая вносит вклад при оценивании популярности пользователя.

Программная реализация

В процессе исследований был разработан программный комплекс, содержащий модули извлечения информации из социальных сетей, обработки, анализа и визуализации данных. Все модули реализованы на языке Python для широкого круга операционных систем, на которых может работать комплекс. Программный комплекс содержит шесть модулей: извлечения данных, обработки данных, отслеживания изменений в пользовательских данных, анализа данных, построения графовых структур, визуализации данных.

Модуль извлечения данных позволяет извлекать данные из Twitter и ВКонтакте. Для доступа к данным в каждой из них используется интерфейс прикладного программирования (API), авторизация производится при помощи протокола OAuth. При использовании одного компьютера программа позволяет в сутки выполнять от 8 до 250 тысяч запросов в зависимости от того, к какой социальной сети осуществляется запрос, и от быстродействия оборудования и пропускной способности каналов. На данный момент модуль извлечения данных имеет возможность функционального расширения практически на любую социальную сеть.

Для хранения извлеченных данных используются документо-ориентированная система управления БД MongoDB и сетевое журналируемое хранилище данных Redis.

Все данные Redis хранит в виде словаря, в котором ключи связаны со своими значениями. Одно из важных отличий Redis от других хранилищ данных заключается в том, что значения этих ключей не ограничиваются строками. Поддерживаются следующие абстрактные типы данных: списки строк, множества строк (коллекции неповторяющихся несортированных элементов), сортированные множества строк (коллекции неповторяющихся элементов), упорядоченных по score (вещественное

число), словари, в которых и ключи, и их значения – строки.

Тип данных значения определяет, какие операции (команды) доступны для него. Хранилище данных Redis поддерживает такие высокоуровневые операции, как объединение и разность наборов, а также их сортировку [11].

Данные с любого сервера Redis могут реплицироваться произвольное количество раз. Репликация – механизм синхронизации содержимого нескольких копий объекта. В процессе репликации изменения, сделанные в одной копии объекта, могут быть распространены в другие копии. Но, чтобы избежать конфликтов, необходимо понимать, в каком порядке должны применяться обновления. При выгрузке информации из социальных сетей приходится иметь дело с очень большими объемами данных. Репликация как раз полезна в таких случаях. Высокая производительность Redis обусловлена тем, что все данные хранятся в оперативной памяти.

Важными условиями при выборе БД были вариативность структуры данных и возможность изменять структуру содержимого БД. Это вызвано тем, что на начальном этапе получалось большое количество неструктурированных данных и сама структура хранения данных постепенно вырабатывалась по мере их извлечения и исследования.

Кроме того, в MongoDB имеются механизмы обработки данных Map Reduce и Aggregation framework, значительно ускоряющие обработку данных в самой БД для предоставления обрабатываемых данных в требуемом формате для последующего анализа. Такой формат организации и хранения данных наиболее подходит для слабоструктурированной информации, в том числе для извлекаемой из социальных сетей.

В реализованном программном комплексе для извлечения данных используются различные параметры запуска в зависимости от социальной сети. Для извлечения данных из сетей Twitter и ВКонтакте используются параметры, приведенные в таблице.

Модуль обработки данных сначала производит поиск хэштегов, упоминаний пользователей и ссылок, далее – нормализацию текста сообщений. Нормализация текста сообщений осуществляется в зависимости от настроек с помощью либо стеммера Портера [12] (для большей скорости обработки), либо морфологической нормализации на основе алгоритмов АОТ [13] с использованием библиотеки PyMorphu.

Модуль слежения за пользовательскими данными позволяет отслеживать изменения, вносимые пользователями в настройки своего аккаунта.

В модуле анализа данных используются различные алгоритмы кластеризации и классификации данных как самих пользователей и их связей, так и их сообщений.

**Описание запускаемых
для социальной сети команд**

**Description of issued instructions
for a social network**

Параметр	Функция
Twitter и ВКонтакте	
-h, --help	Показать справочную информацию о параметрах и выйти из программы
-l LIST, --list LIST	Загрузить данные о нескольких пользователях по списку id пользователей из заранее созданного файла
-u USER, --user USER	Загрузить данные об одном пользователе по имени пользователя
-v, --visualise	Визуализировать данные о пользователях в виде графа
Twitter	
-rt RELATION_TYPE, --relation_type RELATION_TYPE	Указать тип связи: friends или followers; если не указать, по умолчанию берется friends
-d DEPTH, --depth DEPTH	Глубина в цепочке отношений (friends и followers); непосредственные друзья пользователя находятся на глубине 1, друзья друзей – 2 и так далее
ВКонтакте	
-r, --recursive	Загрузить родственные связи пользователя (братья, сестры, родители и пр.)

В модуле построения графовых структур имеется возможность построения графов, отражающих связи пользователей, а также выгрузки данных в программное средство для работы с графами Gephi [14]. Выгрузка может осуществляться в виде csv-файлов или посредством http-протокола. Реализованы оба метода. В зависимости от настроек в Gephi есть возможность отображения вершин, ребер, меток (вершин и ребер), по необходимости можно менять их величину и расцветку, масштабировать изображение с различной степенью детализации, просматривать списки вершин и ребер, ранжировать их. В числе аналитических возможностей Gephi автоматическое вычисление таких характеристик, как диаметр графа, плотность графа, модулярность, средняя длина пути между любыми двумя вершинами, метрики авторитетности вершин HITS и PageRank, центральность по собственному вектору (eigenvector centrality), средний коэффициент кластеризации и пр.

Модуль визуализации данных дает возможность на основе извлеченных данных строить графики зависимостей между различными показателями.

Просмотр результатов работы программы

Для извлечения данных о конкретном пользователе из Twitter необходимо указать имя пользова-

теля и дополнительные параметры, описание которых приведено в таблице. Например, необходимо извлечь информацию о Бахыт Сыздыковой – бывшем депутате мажилиса Республики Казахстан. В Twitter она зарегистрирована под именем @bahytsyzdykova. Часть содержимого коллекции users, в частности, с данными о пользователе @bahytsyzdykova, можно увидеть по ссылке <http://www.swsys.ru/uploaded/image/2015-4-dop/3.jpg>.

У каждого зарегистрированного пользователя ВКонтакте есть собственный ID (идентификационный номер), который присваивается странице во время регистрации. Для запуска извлечения данных о пользователе из сети ВКонтакте необходимо указать этот ID, например 2237709.

Часть содержимого коллекции messages можно увидеть по ссылке <http://www.swsys.ru/uploaded/image/2015-4-dop/4.jpg>.

При помощи запросов к БД можно осуществлять поиск по текстам сообщений, именам пользователей и другим сведениям среди тех данных, которые уже были извлечены. Например, при помощи запроса к БД `db.messages.find({'source': 'vk'})` можно просматривать сообщения, извлеченные только из сети ВКонтакте.

Описание извлекаемых данных

В зависимости от социальной сети можно извлечь четыре типа объектов: User, Message, Social-Object, ContentObject, каждый из которых хранится в своей коллекции в MongoDB. Из Twitter извлекаются только объекты User и Message, из ВКонтакте – все четыре типа.

Объекты типа User характеризуют пользователей социальной сети. Они находятся в коллекции users и имеют различные поля в зависимости от ресурса. Однако для Twitter и ВКонтакте имеются и схожие поля:

source – собственно имя ресурса (ttr – Twitter, vk – ВКонтакте);

name и <[first, last, middle]>_name – имя пользователя, которое он указал при регистрации;

sn_id – идентификатор пользователя в социальной сети;

status – последнее его сообщение (либо его состояние, которое он указал в графе «Что с вами происходит?»);

birthday – дата рождения;

screen_name – ник;

update_date – время сохранения в БД и время последнего просмотра;

[followers, friends]_count – количество подписчиков и друзей пользователя соответственно;

verified – проверенный ли пользователь; в некоторых случаях это ответ на email, посланный при регистрации, в иных – проверка модератором на реальность имени и прочего.

Имеются около 18 видов полей, описывающих пользователя Twitter. Приведем лишь некоторые наиболее важные из них:

`follow_request_sent` – отправлен ли запрос на подписку этому пользователю (true/false);

`time_zone` – временная зона;

`update_date` – время последнего обращения системы к данным этого пользователя;

`protected` – пользователь скрывает свои твиты, твиты показываются только его друзьям и подписчикам;

`listed_count` – количество публичных листов, в которых состоит данный пользователь;

`statuses_count` – количество сообщений пользователя;

`description` – информация, которую пользователь написал о себе;

`favourites_count` – количество избранных объектов у пользователя;

`url` – ссылка, которую указал пользователь в своем описании;

`created_at` – когда был создан аккаунт;

`default_profile` – настроил ли пользователь свой аккаунт (дополнительной индивидуальностью, к примеру, картинкой в фоне);

`following` – читает ли текущий бот искомого пользователя.

Объекты типа `Message` характеризуют сообщения, оставленные пользователем в сети.

1) `status` – последнее сообщение пользователя:

`favorited` – в избранном ли это сообщение у бота;

`text` – текст сообщения;

`favorite_count` – количество пользователей, которые добавили это сообщение себе в избранное;

`lang` – язык текста;

`created_at` – когда создано сообщение;

`retweeted` – было ли переслано сообщение;

`coordinates` – координаты места, откуда было отправлено сообщение;

`source` – ресурс, с которого было создано сообщение;

`retweet_count` – количество пересылок (репостов) сообщения;

`entities` – сущности (которые есть также в сообщениях); все атрибуты этого объекта являются массивами, и каждый элемент является объектом, который содержит информацию о позиции в тексте:

`symbols` : массив символов;

`user_mentions` : массив пользователей, упоминаемых в тексте сообщения;

`hashtags` : массив хэштегов;

`urls` : массив ссылок;

2) `lang` – основной язык пользователя.

Приведем поля для описания объекта типа `User` из ВКонтакте.

1. Поле `city` – выдается id города, указанного у пользователя в разделе «Контакты». Название города по его id можно узнать при помощи метода

`getCities`. Если город не указан, то при приеме данных в формате XML в узле `<user>` отсутствует тег `city`.

2. Поле `country` – выдается id страны, указанной у пользователя в разделе «Контакты». Название страны по ее id можно узнать при помощи метода `getCountries`. Если страна не указана, то при приеме данных в формате XML в узле `<user>` отсутствует тег `country`.

3. Поле `online` – показывает, находится ли этот пользователь сейчас на сайте. Поле доступно только для метода `friends.get`. Возвращаемые значения: 1 – находится, 0 – не находится.

Если пользователь использует мобильное приложение либо мобильную версию сайта, будет возвращено дополнительное поле `online_mobile`.

`lists` – список, содержащий id списков друзей, в которых состоит текущий друг пользователя. Метод получения id и названий списков: `friends.getLists`. Поле доступно только для метода `friends.get`. Если текущий друг не состоит ни в одном списке, то при приеме данных в формате XML в узле отсутствует тег `lists`.

`screen_name` – возвращает короткий адрес страницы (возвращается только имя адреса, например `andrew`). Если пользователь не менял адрес своей страницы, возвращается `'id'+uid`, например `id35828305`.

`has_mobile` – показывает, известен ли номер мобильного телефона пользователя. Возвращаемые значения: 1 – известен, 0 – не известен. Рекомендуется перед вызовом метода `secure.sendSMSNotification`.

`contacts` – возвращает поля:

`mobile_phone` – мобильный телефон пользователя;

`home_phone` – домашний телефон пользователя;

`education` – возвращает код и название университета пользователя, а также факультет и год окончания;

`universities` – список высших учебных заведений, в которых учился текущий пользователь;

`schools` – список школ, в которых учился текущий пользователь.

`can_post` – разрешено ли оставлять записи на стене у данного пользователя.

`can_see_all_posts` – разрешено ли текущему пользователю видеть записи других пользователей на стене данного пользователя.

`can_write_private_message` – разрешено ли написание личных сообщений данному пользователю.

`activity` – возвращает статус, расположенный в профиле, под именем пользователя.

`last_seen` – возвращает объект, содержащий поле `time`, в котором содержится время последнего захода пользователя.

relation – возвращает семейное положение пользователя: 1 – не женат/не замужем; 2 – есть друг/есть подруга; 3 – помолвлен/помолвлена; 4 – женат/замужем; 5 – все сложно; 6 – в активном поиске; 7 – влюблен/влюблена.

4. Поле counters – возвращает количество различных объектов у пользователя. Поле возвращается только в методе getProfiles при запросе информации об одном пользователе. Данное поле является объектом, который содержит следующие поля:

- albums – количество фотоальбомов;
- videos – количество видеозаписей;
- audios – количество аудиозаписей;
- notes – количество заметок;
- friends – количество друзей;
- groups – количество сообществ;
- online_friends – количество друзей онлайн;
- mutual_friends – количество общих друзей (если запрашивается информация не о текущем пользователе);

- user_videos – количество видеозаписей с пользователем;

- followers – количество подписчиков.

Если запрашивается информация не о текущем пользователе, то отсутствие полей friends, online_friends, mutual_friends, user_photos в объекте означает, что информация по ним скрыта соответствующими настройками приватности у запрашиваемого пользователя. Если при запросе данного поля оно отсутствует в ответе, то это означает, что текущий пользователь находится у запрашиваемого пользователя в черном списке.

5. Поле wall_comments – разрешено ли комментирование стены. Если комментирование стены отключено, то комментарии на стене не отображаются.

6. Поле relatives – возвращает список родственников текущего пользователя в виде объектов, содержащих поля uid и type. type может принимать одно из следующих значений: grandchild, grandparent, child, sibling, parent.

7. Поля interests, movies, tv, books, games, about – позволяют получить профильную информацию о пользователе.

8. Поле connections – позволяет получить информацию об аккаунтах пользователя на других сервисах. При указании этого поля будут приходиться следующие ключи в случае наличия соответствующих записей: twitter, facebook, facebook_name, skype, livejournal.

Под объектами типа ContentObject понимаются некоторые контентные объекты, такие как фотографии, видео, обсуждения групп, имеющиеся в сети ВКонтакте. Все эти объекты имеют различные наборы атрибутов в зависимости от типа объекта, к каждому из них могут привязываться объекты типа Message посредством информации из атрибута comment_for.

Также для каждого такого объекта определены следующие поля:

- sn_id – уникальный идентификатор записи;

- user – ссылка на пользователя, загрузившего эту запись;

- text – что пользователь написал про эту запись;

- created_at – когда была создана запись;

- views_count – количество просмотров в некотором случае;

- comments_count – количество комментариев;

- likes_count – количество лайков;

- video_id – идентификаторы видеоконтента в сети vk;

- photo_id – идентификаторы фото контента в сети vk;

- wall_post_id – идентификаторы записей на стене пользователя vk;

- type – тип [wall_post, video, photo].

Объекты типа SocialObject представляют сообщества пользователей, такие как группы, публичные страницы, встречи, и различаются свойством type.

Для каждого объекта SocialObject определены следующие поля:

- sn_id – идентификатор группы, страницы, встречи в социальной сети vk; в некоторых случаях уместно его применение в обратном значении, таком как get_photos;

- private – закрыто ли это сообщество для сторонних пользователей (чтобы вступить, требуется отправить запрос администратору этого сообщества);

- name – видимое всем пользователям имя сообщества;

- screen_name – «ник» сообщества;

- type – тип сообщества (может быть group, page, event);

- source – всегда равен vk.

Связи между двумя пользователями либо между пользователем и сообществом (для социальной сети ВКонтакте) ввиду их большого количества хранятся в Redis для последующего быстрого извлечения. Каждая связь имеет определенный тип и соединяет между собой либо двух пользователей, либо пользователя и сообщество. Таким образом, связь – это тройка $L = \langle B, U, T \rangle$, где B – идентификатор начального пользователя; U – идентификатор конечного пользователя (либо сообщества, то есть группы пользователей); T – тип связи.

Связи между пользователями в социальной сети Twitter имеют только 2 типа: friend и follower. Связи между двумя пользователями социальной сети ВКонтакте могут быть следующими: friend, follower, like, comment, mentions. Типы связей между пользователем и сообществом: member, admin, subscribe, request, invitation.

На основании изложенного сделаем следующие выводы. В процессе анализа социальных сетей решается довольно большой круг задач и применя-

ются методы из различных областей знаний. Часто решение задач одного класса связано с решением задач другого класса, поэтому к их решению целесообразно подходить комплексно. Для этого, безусловно, необходимы новые методы, алгоритмы, введение новых характеристик, которые помогли бы в решении возникающих вопросов.

В рамках исследования был разработан программный комплекс для анализа данных из социальных сетей. Он состоит из шести модулей: извлечения, обработки данных, отслеживания изменений пользовательских данных, анализа данных, построения графовых структур и модуля визуализации данных. В данной работе также приведено довольно подробное описание извлекаемых данных.

Ясно, что количество получаемой информации очень большое. При использовании распределенной системы извлечения и обработки данных объем возрастает еще больше. Поэтому сначала необходимо из всей информации выделить ту, которую можно было бы достаточно эффективно обработать и которая представляла бы интерес в соответствии с поставленными целями. К примеру, необходимо выяснить, насколько большое влияние оказывает на человека его окружение. Для решения этой задачи предлагается применить к социальным сетям теорию динамического социального влияния, предложенную Латане.

Задача определения количества влияния на пользователя со стороны других пользователей непосредственно связана с задачей обнаружения лидеров мнений – относительно популярных пользователей, формирующих мнение большинства. Благодаря им информация распространяется быстрее и, соответственно, воздействует на более многочисленную аудиторию. Лидера мнений характеризуют следующие величины: количество подписчиков или друзей, количество созданных сообщений, количество лайков, репостов и комментариев, которые собирают записи пользователя. Эти характеристики позволяют более точно оценить популярность пользователя.

Проведенные испытания созданного программного комплекса показали его корректную работу. Извлекаемые из сетей данные весьма интересны. Однако необходимы еще более глубокое осмысление получаемых результатов и дополнительные исследования для того, чтобы с помощью программного комплекса получать и анализировать действительно наиболее важную информацию.

Литература

1. Twitter. URL: <https://en.wikipedia.org/wiki/Twitter> (дата обращения: 1.07.2015).
2. Nowak A., Szamrej J., Latané B. From private attitude to public opinion: a dynamic theory of social impact. *Psychological Review*, 1990, vol. 97, pp. 362–376.
3. Latané B. The psychology of social impact. *American Psychologist*, 1981, vol. 36, pp. 343–356.
4. Роджерс Э., Агарвала-Роджерс Р. Коммуникации в организациях. М.: Экономика, 1980. 178 с.
5. Крючков В.Н., Мурзин Ф.А., Нартов Б.К. Исследование связей в коллективах и сетях ЭВМ на основе анализа предпочтений // Проблемы конструирования эффективных и надежных программ. 1995. С. 136–141.
6. Number of monthly active VKontakte users via desktop connections in Russia from December 2012 to December 2014. The Statistics Portal. URL: <http://www.statista.com/statistics/425423/number-of-monthly-active-vkontakte-users> (дата обращения: 1.07.2015).
7. Top social networks in Russia: latest numbers and trends. Russian Search. URL: <http://www.russiansearchtips.com/2015/01/top-social-networks-russia-latest-numbers-trends> (дата обращения: 1.07.2015).
8. Сколько тратят казахстанские компании на рекламу в соцсетях. Forbes Kazakhstan. URL: http://forbes.kz/process/internet/skolko_tratyat_kazahstanskije_kompanii_na_reklamu_v_sotssetyah (дата обращения: 1.07.2015).
9. Number of monthly active international Twitter users from 2nd quarter 2010 to 1st quarter 2015. The Statistics Portal. URL: <http://www.statista.com/statistics/274565/monthly-active-international-twitter-users> (дата обращения: 1.07.2015).
10. Hirsch J.E. An index to quantify an individual's scientific research output. *Proc. Natl Acad. Sci. USA*. 2005, no. 102 (46), pp. 16569–16572.
11. Redis. URL: <https://ru.wikipedia.org/wiki/Redis> (дата обращения: 1.07.2015).
12. Willett P. The Porter stemming algorithm: then and now. *Program: Electronic Library and Information Systems*, 2006, vol. 40, no. 3, pp. 219–223.
13. Автоматическая обработка текста. URL: <http://aot.ru> (дата обращения: 1.07.2015).
14. Gephi, an open source graph visualization and manipulation software. URL: <https://gephi.org> (дата обращения: 1.07.2015).

DOI: 10.15827/0236-235X.112.188-197

Received 02.07.15

A SOFTWARE PACKAGE FOR A SOCIAL NETWORK DATA ANALYSIS

Batura T.V., Ph.D. (Physics and Mathematics), Senior Researcher, tatiana.v.batura@gmail.com;

Murzin F.A., Ph.D. (Physics and Mathematics), Deputy Director on Scientific Work, murzin@iis.nsk.su;

Proskuryakov A.V., Postgraduate Student, alexey.proskuryakov@gmail.com

(A.P. Ershov Institute of Informatics Systems (IIS), Siberian Branch of the Russian Federation Academy of Sciences, Lavrenteva Av. 6, Novosibirsk, 630090, Russian Federation)

Abstract. The paper focuses on the problems of extracting and processing social network data. It considers various numerical characteristics, relations and sets analyzed on the basis of output data. It is important that these characteristics are constructive and can be effectively calculated or built using corresponding algorithms.

The data amount in social networks is large. It becomes even larger when using distributed data extraction and processing system. That is why the most significant and difficult problem is to determine data, which could be processed effectively enough and is of interest according to the goals. To solve this task, we suggest using the method of determining the amount of

audience influence on individual users. The method is based on the dynamic social impact theory proposed by B. Latané. This approach is also useful for solving the problem of determining the primary source of information. This problem is connected to the opinion leader detection problem. Opinion leaders are popular users who form the opinion of the majority. The paper describes one of the methods of opinion leader detection.

The article provides a detailed description of the developed software, which processes information from social networks VKontakte and Twitter. It consists of six modules: extraction module, data processing module, user data change tracking module, data analysis module, graph construction module and data visualization module.

Keywords: social networks analysis, natural language data processing, primary source of information dissemination, opinion leaders, interpersonal analysis.

References

1. Twitter. 2015. Available at: <https://en.wikipedia.org/wiki/Twitter> (accessed July 1, 2015).
2. Nowak A., Szamrej J., Latané B. From private attitude to public opinion: a dynamic theory of social impact. *Psychological Review*. 1990, vol. 97, pp. 362–376.
3. Latané B. The psychology of social impact. *American Psychologist*. 1981, vol. 36, pp. 343–356.
4. Rogers E.M., Agarwala-Rogers R. *Communication in organizations*. NY, Free Press, 1976, 209 p. (in Russ. ed.: Moscow, Ekonomika Publ., 1980, 178 p.).
5. Kryuchkov V.N., Murzin F.A., Nartov B.K. An Investigation of the Connections in the Collectives and in the Computer Networks. *Problemy konstruirovaniya effektivnykh i nadezhnykh programm* [The Problems of Constructing Efficient and Reliable Programs]. Novosibirsk, 1995, pp. 136–141 (in Russ.).
6. Number of monthly active VKontakte users via desktop connections in Russia from December 2012 to December 2014. *The Statistics Portal*. 2015. Available at: <http://www.statista.com/statistics/425423/number-of-monthly-active-vkontakte-users> (accessed July 1, 2015).
7. Top social networks in Russia: latest numbers and trends. *Russian Search Tips*. 2015. Available at: <http://www.russiansearchtips.com/2015/01/top-social-networks-russia-latest-numbers-trends> (accessed July 1, 2015).
8. How much do Kazakhstan companies spend on advertisement in social networks? *Forbes Kazakhstan*. 2015. Available at: http://forbes.kz/process/internet/skolko_tratyat_kazahstanskije_kompanii_na_reklamu_v_sotssetyah (accessed July 1, 2015).
9. Number of monthly active international Twitter users from 2nd quarter 2010 to 1st quarter 2015. *The Statistics Portal*. 2015. Available at: <http://www.statista.com/statistics/274565/monthly-active-international-twitter-users> (accessed July 1, 2015).
10. Hirsch J.E. An index to quantify an individual's scientific research output. *Proc. National Acad. Sci. USA*. 2005, no. 102 (46), pp. 16569–16572.
11. Redis. 2015. Available at: <https://ru.wikipedia.org/wiki/Redis> (accessed July 1, 2015).
12. Willett P. The Porter stemming algorithm: then and now. *Program: Electronic Library and Information Systems*. 2006, vol. 40, no. 3, pp. 219–223.
13. *Avtomaticheskaya obrabotka teksta* [Automatic Text Processing]. 2012. Available at: <http://aot.ru/> (accessed July 1, 2015).
14. Gephi, an open source graph visualization and manipulation software. 2012. Available at: <https://gephi.org> (accessed July 1, 2015).

Примеры оформления статьи в списке литературы

1. Батура Т.В., Мурзин Ф.А., Проскуряков А.В. Программный комплекс для анализа данных из социальных сетей // Программные продукты и системы. 2015. № 4 (112). С. 188–197.

2. Батура Т.В., Мурзин Ф.А., Проскуряков А.В. Программный комплекс для анализа данных из социальных сетей // Программные продукты и системы. 2015; DOI: 10.15827/0236-235X.112. 188-197.

3. Batura T.V., Murzin F.A., Proskuryakov A.V. A software package for a social network data analysis. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 4 (112), pp. 188–197 (in Russ.); DOI: 10.15827/0236-235X.112.188-197.

РЕПРЕЗЕНТАТИВНОСТЬ МЕТРИК НА ОСНОВЕ СОБЫТИЙ ПРОЦЕССОРА INTEL SANDY BRIDGE ПРИ АНАЛИЗЕ ВРЕМЕНИ ОБРАБОТКИ ДАННЫХ В ПАМЯТИ

Е.Ю. Иванов, аспирант, i@eivanov.com;

*М.С. Косяков, к.т.н., доцент, mkosyakov@gmail.com, mkosyakov@tbricks.com
(Санкт-Петербургский национальный исследовательский университет информационных
технологий, механики и оптики (Университет ИТМО),
Кронверкский просп., 49, г. Санкт-Петербург, 197101, Россия)*

В данной работе выполнена оценка репрезентативности метрик на основе событий процессора Intel Sandy Bridge при анализе времени обработки данных в памяти. Приведены детальный обзор и анализ наиболее часто используемых при профилировании метрик, рекомендованных в документации Intel. Разработан синтетический тест, позволяющий измерить временные задержки при последовательном, случайном и постраничном доступах к памяти, различных размерах всего рабочего множества и отдельных его элементов, обрабатываемых тестовым приложением. Описан способ профилирования и получения числа событий процессора, основанный на использовании утилиты perf ОС Linux. С помощью теста получены временные характеристики работы с оперативной памятью и рассчитаны метрики на основе событий процессора. Экспериментально установлено, что при последовательном доступе к данным зависимость рекомендованных компанией Intel метрик CPI, SCPI, PSRC и частоты кэш-промахов от измеренных временных задержек имеет линейный характер. Кроме того, изменение данных метрик пропорционально изменению времени с коэффициентом, близким к единице. На основе полученных результатов для случайного доступа к памяти показано, что наиболее эффективной метрикой является число тактов, затраченных на ожидание шины данных, так как данная метрика хорошо описывает временные задержки в работе приложений, выполняющих обработку данных вне зависимости от их расположения в памяти. По мнению авторов, указанная метрика наиболее удобна для оценки эффективности оптимизации работы приложения с оперативной памятью. На основе полученных результатов показано, что рассматриваемые в работе метрики являются репрезентативными и могут быть использованы при анализе временных характеристик приложений, обрабатывающих данные в оперативной памяти компьютера.

Ключевые слова: *счетчики производительности, события процессора, временные задержки, доступ к памяти, профилирование, оптимизация, быстроедействие, кэш-память, intel sandy bridge.*

При разработке ПО важным этапом оптимизации времени выполнения является профилирование. Процесс профилирования заключается в выявлении основных характеристик работы приложения, таких как наиболее часто выполняемые участки исходного кода и точки вызова системных функций, время работы отдельных функций, эффективность использования ресурсов процессора. Результаты профилирования позволяют найти исходный код, оказывающий наибольшее влияние на общее время выполнения программы или отдельных библиотечных функций, что помогает оптимизировать приложения. Некоторые типы профилирования могут оказывать влияние на быстродействие приложения, однако часто данное влияние можно свести к минимуму [1].

Для оценки эффективности использования ресурсов процессора обычно применяются метрики на основе его событий. Современные процессоры Intel предоставляют пользователю не менее 8 физических счетчиков производительности (от англ. performance counters) [2], которые можно запрограммировать на подсчет практически любых событий процессора, например кэш-промахов или числа выполненных микроопераций [3, 4]. Метрики на основе числа событий процессора позволяют обнаружить исходный код, неэффективный с точки зрения использования ресурсов процессора, причиной чего могут быть проблемы как в реализации на вы-

сокоуровневом языке, так и при генерации машинных инструкций компилятором. При использовании таких метрик основная сложность заключается в сопоставлении их значений со временем выполнения профилируемого кода. Проанализируем репрезентативность наиболее часто используемых метрик на основе событий процессора Intel Sandy Bridge и временных характеристик работы приложений, выполняющих обработку данных, хранящихся в оперативной памяти компьютера. Результаты данного анализа могут быть использованы для оценки изменения времени обработки данных из оперативной памяти после оптимизации приложения исключительно по метрикам на основе счетчиков событий процессора.

Обзор основных метрик на основе числа событий процессора

Для общей оценки эффективности использования ресурсов процессора компания Intel в своей документации для инструмента Intel VTune Amplifier XE [5] предлагает следующие метрики:

- CPI (cycles per instruction) – среднее число тактов процессора, затраченное на выполнение одной инструкции;
- SCPI (stalled cycles per instructions) – среднее число холостых тактов процессора на одну инструкцию;

– PSRC (pipeline slots retired per cycle) – среднее число освобожденных слотов конвейера процессора за один такт.

CPI вычисляется как отношение числа следующих событий процессора: CPU_CLK_UNHALTED.THREAD / INST_RETIRED.ANY (здесь и далее мы используем названия событий процессора из документации Intel), где INST_RETIRED.ANY – число выполненных инструкций, а CPU_CLK_UNHALTED.THREAD – число тактов процессора, затраченных на их выполнение. Обычно референсные значения CPI измеряются с помощью различных синтетических тестов, после чего они используются при сравнении со значениями, полученными для профилируемого приложения. Благодаря параллелизму уровня инструкций значение CPI в общем случае должно быть меньше или равно единице. Однако это значение сильно зависит от нагрузки, поэтому обычно CPI используется для сравнения эффективности каждого шага оптимизации: уменьшение CPI свидетельствует об улучшении быстродействия, увеличение – о его ухудшении.

SCPI часто применяется для оценки накладных расходов на доступ к оперативной памяти и рассчитывается как отношение значений счетчика UOPS_ISSUED.ANY, запрограммированного для подсчета тактов процессора, во время которых выполнение инструкций было заблокировано (от англ. stalled cycles), к счетчику INST_RETIRED.ANY. Чем выше значение SCPI, тем больше временные задержки, поэтому при оптимизации приложения разработчики должны стремиться к снижению значения SCPI.

Метрика PSRC показывает, насколько эффективен конвейер процессора. Согласно документации Intel, метрика рассчитывается по следующей формуле: $UOPS_RETIRED.RETIRE_SLOTS / (4 * CPU_CLK_UNHALTED.THREAD)$, где UOPS.RETIRED.RETIRE_SLOTS – общее число выполненных микроопераций. Множитель 4 обусловлен тем, что в процессорах Intel на базе микроархитектуры Core второго поколения, куда относятся процессоры Intel Sandy Bridge, за один такт может быть выполнено до 4 микроопераций. Более высокие значения PSRC свидетельствуют о более эффективном использовании процессора.

В дополнение к метрикам, описанным выше, оценим влияние частоты кэш-промахов и числа тактов, затраченных на доступ к шине данных (от англ. bus cycles) на время выполнения приложений, обрабатывающих данные, хранящиеся в оперативной памяти.

Временные задержки при доступе к памяти на процессорах Intel Sandy Bridge

Для измерения временных характеристик работы приложений при различных типах доступа к памяти, размерах рабочего множества (от англ.

working set) и обрабатываемых данных авторами на основе популярных синтетических тестов [6] разработано тестовое приложение на языке C++. В приложении создается список из элементов, размер которых можно варьировать, и выполняется обход всех элементов списка. Число элементов в списке выбирается таким образом, чтобы размер списка соответствовал размеру рабочего множества, указанного при запуске приложения. Также при запуске указывается порядок обхода элементов: последовательный (элементы расположены в памяти последовательно), случайный или постраничный (каждый элемент расположен на своей странице виртуальной памяти). В случае, когда каждый элемент расположен на своей странице памяти, смещение к элементу списка внутри страницы случайно, что позволяет избежать так называемых ассоциативных промахов (от англ. associativity miss), вызванных особенностью кэш-памяти с множественным доступом [7]. Важно отметить, что при обходе считываются только первые 8 байт элемента, в которых хранится адрес следующего элемента. Результатом выполнения является среднее время t , затраченное на посещение одного элемента списка, измеренное в тактах процессора.

Эксперименты выполнены на сервере с процессором Intel Xeon E5-2690 с тактовой частотой 2.9 GHz. С целью повышения точности результатов функция Hyper-Threading процессора отключена. Сервер работает под управлением операционной системы Oracle Linux 6.5 с ядром linux-3.8, размер страницы виртуальной памяти равен 4 Кб. Intel Xeon E5-2690 имеет следующие параметры:

- размер кэша данных первого уровня – 32 Кб;
- размер кэша второго уровня – 256 Кб;
- размер кэша третьего уровня – 20 480 Кб;
- длина кэш-линейки – 64 байта;
- число записей DTLB первого уровня – 64 (для страниц виртуальной памяти размером 4 Кб);
- число записей TLB второго уровня – 512 (для страниц виртуальной памяти размером 4 Кб).

На рисунке 1 приведены графики зависимости среднего времени t посещения одного элемента списка от размера рабочего множества для элементов размером 8, 64 и 256 байт. Для списка из восьмибайтных элементов t не зависит от размера рабочего множества и размеров кэшей различного уровня, что объясняется крайне высокой эффективностью упреждающей выборки из оперативной памяти (от англ. memory prefetch) [8]. В случае, когда размер элемента равен 64 байтам, каждый элемент списка занимает свою кэш-линейку, поэтому при посещении каждого элемента необходимо загрузить запрашиваемую кэш-линейку, что не позволяет выполнять упреждающую выборку. Таким образом, на графике четко видны задержки при доступе к различным уровням кэша, обусловленные соответствующими кэш-промахами. Когда размер

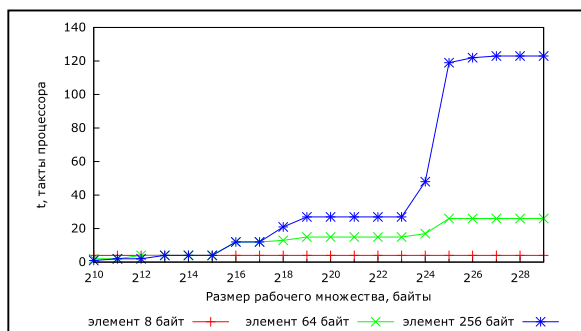


Рис. 1. Среднее время t обработки одного элемента списка в зависимости от размера элемента и размера рабочего множества

Fig. 1. Average time t for processing one element from the list depending on the size of this element and working set

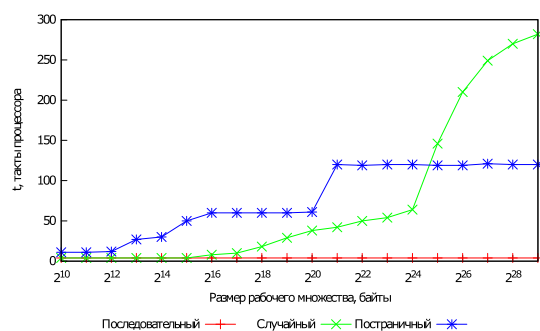


Рис. 2. Среднее время t обработки одного элемента размером 8 байт списка в зависимости от размера рабочего множества и порядка обхода элементов

Fig. 2. Average time t for processing one 8-bit element from the list depending on the size of working set and winding order

элемента равен 256 байтам, увеличение t по сравнению с обходом элементов размером 64 байта обусловлено значительными накладными расходами на промахи в кэш DTLB, что будет показано далее.

Для демонстрации влияния промахов кэша DTLB на рисунке 2 приведены графики для различных порядков обхода списка, состоящего из элементов размером 8 байт. Хорошо видно, что график для случая, когда каждый элемент расположен на своей странице виртуальной памяти, практически совпадает с графиком для последовательного об-

хода элементов размером 256 байт. Но для списка с элементами размером 256 байт время t незначительно ниже, так как перед каждым промахом DTLB выполняется посещение 8 элементов, благодаря чему промахи DTLB случаются не так часто. При случайном порядке обхода промахи DTLB случаются реже, чем при постраничном, однако следует ожидать большого числа кэш-промахов, чем обусловлено значительно более высокое время t по сравнению с последовательным обходом [9]. Кроме того, при случайном доступе снижается быстродействие оперативной памяти [10]. Аналогичные результаты получены рядом исследователей для других современных процессоров с архитектурой x86_64 [6, 7, 11].

Анализ репрезентативности метрик при оценке изменения временных задержек

Для получения значений числа событий процессора и дальнейшего расчета метрик выполнено профилирование тестовой программы, описанной выше. Для профилирования использована утилита perf [12], которая позволяет активировать счетчики производительности во время выполнения программы для заданных событий процессора и по ее завершении получить их значения. В таблице приведены нормализованные значения времени t и значения числа событий. Результаты для случайного обхода включены в таблицу, но рассматриваются далее отдельно.

Согласно полученным результатам, при последовательном и постраничном обходах (последовательном доступе к памяти) зависимость между средним временем t обработки одного элемента и всеми рассматриваемыми метриками имеет линейный характер. Более того, различия между нормализованными значениями находятся в пределах вычислительной погрешности, то есть можно считать их идентичными. Таким образом, в случае, когда обращение к данным осуществляется последовательно, при оптимизации программы следует ожидать изменения задержек времени во столько же раз, во сколько раз изменились значения метрик CPI, SCPI, PSRC, частоты кэш-промахов или числа тактов, затраченных на ожидание шины данных. При случайном доступе к памяти наиболее эффек-

Нормализованные значения среднего времени t обработки одного элемента списка и исследуемых метрик при различных порядках обхода и размерах элемента

Average time normalized values t for processing one element from the list and analyzed metrics for different winding orders and element size

Порядок обхода, размер элемента в байтах	T	CPI	SCPI	PSRC	Частота кэш-промахов	Такты на доступ к шине данных
Последовательный, 8	1	1	1	1	1	1
Последовательный, 64	5	6,25	6	7	12	4
Последовательный, 256	28	25	27	28	67	23
Постраничный, 8	84	100	92	46	72,5	71
Случайный, 8	46	20	18	14	21	42

тивной метрикой является число тактов, затраченных на ожидание шины данных: данная метрика всегда имеет высокую степень репрезентативности временных характеристик работы приложений, выполняющих обработку данных, вне зависимости от их расположения в памяти.

В заключение отметим, что в данной работе выполнена оценка репрезентативности метрик на основе событий процессора Intel Sandy Bridge при анализе временных характеристик приложений, обрабатывающих данные в оперативной памяти компьютера. Для этого авторами разработано тестовое приложение, позволяющее измерить временные задержки при последовательном, случайном и постраничном доступе к памяти, различных размерах всего рабочего множества и отдельных его элементов, обрабатываемых тестовым приложением. Последовательный, случайный и постраничный обходы элементов рабочего множества соответствуют различным вариантам доступа к произвольным данным в оперативной памяти. С целью получения числа событий процессора и расчета метрик на их основе с помощью утилиты *perf* операционной системы Linux выполнено профилирование тестового приложения.

Установлено, что для программ, выполняющих обработку данных, хранящихся в оперативной памяти компьютера, при последовательном доступе к данным зависимость метрик CPI, SCPI, PSRC и частоты кэш-промахов от измеренных временных задержек имеет линейный характер. Более того, коэффициент пропорциональности данных метрик близок к единице. При случайном доступе к памяти наиболее эффективной метрикой является число тактов, затраченных на ожидание шины данных: данная метрика хорошо описывает временные задержки в работе приложений, выполняющих обработку данных, вне зависимости от их расположения в памяти.

Литература

1. Иванов Е.Ю., Торопов А.В., Косяков М.С. Исследование влияния профилирования памяти средствами библиотеки *jemalloc* на время выполнения многопоточных приложений // Программные продукты и системы. 2015. № 2. С. 55–59.
2. Gregg B. *Systems Performance: Enterprise and the Cloud*. Pearson Education, 2013, 792 p.
3. Intel 64 and IA-32 Architectures Software Developer's Manual, vol. 3B: System Programming Guide, Intel, Inc. 2015, Part 2, 554 p. URL: <http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-3b-part-2-manual.pdf> (дата обращения: 25.06.2015).
4. Intel 64 and IA-32 Architectures Optimization Reference Manual. Intel, Inc. 2014, 642 p. URL: <http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architecture-optimization-manual.html> (дата обращения: 25.06.2015).
5. Using Intel VTune Performance Analyzer to Optimize Software for the Intel Core i7 Processor Family. Intel, Inc., 2010. URL: <https://software.intel.com/en-us/articles/using-intel-vtune-performance-analyzer-to-optimize-software-for-the-intel-core-i7-processor-family> (дата обращения: 25.06.2015).
6. Drepper U. What every programmer should know about memory. Red Hat, Inc. 2007, vol. 11, 114 p. URL: <https://people.freebsd.org/~lstewart/articles/cpumemory.pdf> (дата обращения: 25.06.2015).
7. Ballabriga C., Casse H. Improving the first-miss computation in set-associative instruction caches. Real-Time Systems, 2008. ECRTS'08. Euromicro Conference on. IEEE, 2008, pp. 341–350.
8. Peng L. et al. Memory performance and scalability of Intel's and AMD's dual-core processors: a case study. Performance, Computing, and Communications Conf., 2007 (IPCCC 2007). IEEE International. IEEE, 2007, pp. 55–64.
9. Zhong Y. et al. Miss rate prediction across program inputs and cache configurations. Computers, IEEE Transactions on, 2007, vol. 56, no. 3, pp. 328–343.
10. Солнушкин К.С. Анализ быстродействия ОЗУ и построение модели производительности ЭВМ // Науч.-технич. вестн. информ. технологий, механики и оптики. 2008. № 9 (54). С. 180–182.
11. Molka D. et al. Memory performance and cache coherency effects on an Intel Nehalem multiprocessor system. Parallel Architectures and Compilation Techniques, 2009. PACT'09. 18th International Conf. on. IEEE, 2009, pp. 261–270.
12. Утилита Linux *perf*. URL: https://perf.wiki.kernel.org/index.php/Main_Page, свободный. Яз. англ. (дата обращения: 05.10.2014).

DOI: 10.15827/0236-235X.112.198-202

Received 29.06.15

REPRESENTATIVENESS OF INTEL SANDY BRIDGE PERFORMANCE COUNTERS BASED METRICS FOR MEMORY ACCESS LATENCY ANALYSIS

Ivanov E. Yu., Postgraduate Student, i@eivanov.com;

Kosyakov M.S., Ph.D. (Engineering), Associate Professor, mkosyakov@gmail.com, mkosyakov@tbricks.com
(The National Research University of Information Technologies, Mechanics and Optics,
Kronverksky Ave. 49, St. Petersburg, 197101, Russian Federation)

Abstract. This paper examines the representativeness of metrics based on Intel Sandy Bridge performance event counters and memory access latencies for data processed in RAM. It contains detailed overview and analyses of frequently used metrics recommended by Intel documentation. The authors have implemented a synthetic test application, which allows measuring memory access latencies for serial, per-page and random memory accesses and different sizes of working set and its elements processed by test application. The paper describes an approach to access performance event counters based on Linux *perf* utility. The test presented memory access latencies and calculated metrics based on performance event counters. The experimental studies showed that CPI, SCPI and PSRC metrics recommended by Intel and cache-miss ratio with serial memory access have linear dependency with measured memory latencies. Besides, the metrics values change proportionally to the change of memory latencies with the coefficient close to one. The random access results showed that most efficient metric for performance is the number of cycles wasted on bus waiting for data, because it always has high representativeness of memory access latencies and does not depend on data location. According to authors' analyses, this metric is the most useful to evaluate

memory access performance optimization efficiency. The results showed that the metrics considered in this paper can be used to analyze memory characteristics of application, which process data in RAM.

Keywords: performance event counters, memory access latencies, optimization, cache-memory, Intel Sandy Bridge.

References

1. Ivanov E.Yu., Toropov A.V., Kosyakov M.S. An experimental study of the performance impact of jemalloc memory profiler on execution time of multithreaded applications. *Programmnye produkty i sistemy* [Software & Systems]. 2015, no. 2, pp. 55–59 (in Russ.).
2. Gregg B. *Systems Performance: Enterprise and the Cloud*. Pearson Education. 2013, 792 p.
3. Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide. Part 2, 2014. Available at: <http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-3b-part-2-manual.pdf> (accessed June 25, 2015).
4. Intel 64 and IA-32 Architectures Optimization Reference Manual. 2014. Available at: <http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html> (accessed June 25, 2015).
5. Using Intel VTune Performance Analyzer to Optimize Software for the Intel Core i7 Processor Family. 2010. Available at: <https://software.intel.com/en-us/articles/using-intel-vtune-performance-analyzer-to-optimize-software-for-the-intel-core-i7-processor-family> (accessed June 25, 2015).
6. Drepper U. What every programmer should know about memory. *Red Hat, Inc.* 2007, vol. 11. Available at: <https://people.freebsd.org/~lstewart/articles/cpumemory.pdf> (accessed June 25, 2015).
7. Ballabriga C., Casse H. Improving the first-miss computation in set-associative instruction caches. *ECRTS'08. Euromicro Conf. on Real-Time Systems. IEEE*. 2008, pp. 341–350.
8. Peng L. Memory performance and scalability of Intel's and AMD's dual-core processors: a case study. *Performance, Computing, and Communications Conf., 2007. IPCCC 2007. IEEE Int.* 2007, pp. 55–64.
9. Zhong Y. et al. Miss rate prediction across program inputs and cache configurations. *IEEE Transactions on Computers*. 2007, vol. 56, no. 3, pp. 328–343.
10. Solnushkin K.S. Main memory speed analysis and pc performance modeling. *Nauchno-tehnichesky vestnik informatsionnykh tekhnology, mekhaniki i optiki* [Scientific and Technical Journ. of Information Technologies, Mechanics and Optics]. 2008, no. 9 (54), pp. 180–182 (in Russ.).
11. Molka D. Memory performance and cache coherency effects on an Intel Nehalem multiprocessor system. *IEEE 18th Int. Conf. on Parallel Architectures and Compilation Techniques (PACT'09)*. 2009, pp. 261–270.
12. *Linux perf*. Available at: https://perf.wiki.kernel.org/index.php/Main_Page (accessed June 15, 2015).

Реклама

НИИ «Центрпрограммсистем»

- Крупнейшая IT-компания с 45-летней историей.
- Более чем 20-летний опыт работы в области защиты информации.
- Собственная лаборатория, отвечающая всем нормативным требованиям.
- Наличие всех необходимых сертификатов и лицензий.
- Современная контрольно-измерительная аппаратура, включая сертифицированные ФСТЭК России автоматизированные комплексы.

Полный комплекс работ по защите государственной тайны:

- аттестационные испытания на соответствие требованиям безопасности с выдачей аттестата соответствия;
- специальные проверки для выявления технических средств перехвата информации (закладных устройств);
- специальные исследования для выявления возможных каналов утечки информации посредством побочных электромагнитных излучений и наводок.

Подробности на сайте www.cpstver.ru

Телефон в Москве: +7 (495) 720-59-00

Телефон в Твери: (4822) 69-06-23

УДК 004.652.4: 004.7

Дата подачи статьи: 01.07.15

DOI: 10.15827/0236-235X.112.203-209

РЕПЛИКАЦИЯ ДАННЫХ В ИЕРАРХИЧЕСКИХ ИНФОРМАЦИОННЫХ СИСТЕМАХ С НЕПОСТОЯННОЙ СВЯЗНОСТЬЮ УЗЛОВ

*В.Е. Сорокин, к.т.н., ведущий научный сотрудник, sorokinve@yandex.ru
(НИИ «Центрпрограммсистем», пр. 50 лет Октября, 3а, г. Тверь, 170024, Россия)*

Особенность многих иерархических информационных систем, ядром которых являются размещенные в их узлах реляционные БД, заключается в непостоянстве связи между узлами. Она препятствует классическому использованию централизованной или распределенной БД, возлагая на специальное ПО, решающее функциональные задачи системы, дополнительные функции обмена специализированными сообщениями и консолидации данных, в том числе от внешних для информационных систем источников. Репликация в локальной БД необходимых фрагментов БД других узлов информационных систем, консолидируемая с информацией из внешних источников, позволяет значительно упростить и повысить унификацию специального ПО различных узлов информационных систем. В статье предлагается подход к преобразованию исходной структуры БД, основанный на создании отдельных для различных источников данных таблиц, наследованных от исходных, адаптированный к особенностям асинхронной master/slave-репликации иерархически связанных БД. Показана возможность использования управляемых триггерами флагов и процедур для реализации различных стратегий консолидации данных и простой адаптации исходного специального ПО для решения функциональных задач информационных систем на преобразованной БД. Приводится пример реализации описанного подхода для широко распространенной и используемой во многих отечественных сертифицированных программных средствах объектно-реляционной СУБД PostgreSQL с системой асинхронной master/slave-репликации Slony-I. Отмечается возможность преодоления известных ограничений использования данной системы асинхронной master/slave-репликации применением модулей dblink поддержки сессионного соединения с другими БД и postgres_fdw обертки внешних данных, а также совместного использования системы репликации с этими модулями на преобразованной БД.

Ключевые слова: иерархическая информационная система, реляционная БД, асинхронная репликация, наследование таблиц, обертка внешних данных.

Иерархические информационные системы (ИС) относятся к одним из наиболее распространенных. Их ядром во многих случаях являются реляционные БД, и для решения возложенных на ИС задач необходим доступ к данным не только своего, но и других узлов. При этом классическими решениями являются система «клиент/сервер» с единой централизованной БД или распределенная по узлам ИС БД. В обоих случаях требуется постоянный надежный доступ к другим узлам ИС, что далеко не всегда обеспечивается.

Наглядным примером могут служить иерархические ИС с ситуационными центрами во главе структуры, на которые замыкаются стационарные региональные узлы с подчиненными им мобильными узлами, что характерно для управления территориальными надзорными и ремонтными работами или устранения последствий чрезвычайных ситуаций. При этом основными узлами принятия общих решений, формирования итоговых документов и источником директивной информации являются ситуационные центры, а источником первичной информации – мобильные узлы, связь с которыми может быть непостоянной. В отсутствии актуальной первичной информации для принятия решений, формирования документов и т.п. используется информация из внешних для ИС источников, что дополнительно усложняет консолидацию (непротиворечивое слияние) данных, поступивших из нескольких разных источников.

Исторически типичным в таких случаях является решение *специальным ПО* (СПО) функцио-

нальных задач ИС как во взаимодействии с локальной БД, так и при вводимых оператором внешних данных и обмене специализированными сообщениями между СПО, размещенным в различных узлах ИС. Несмотря на то, что значительная часть задач, решаемых в различных узлах ИС, идентична или подобна, обмен сообщениями между СПО существенно препятствует его унификации.

Независимая от СПО репликация в локальной БД необходимых фрагментов БД других узлов ИС, консолидируемая с информацией из внешних источников, позволяет значительно упростить и повысить унификацию СПО различных узлов ИС. Одному из практических подходов к репликации фрагментов реляционных БД в иерархических ИС с непостоянной связностью узлов посвящена данная статья.

Сравнение возможных способов репликации реляционных БД

Теоретически строго полноценной и гарантирующей целостность БД является синхронная репликация, при которой после обновления одной реплики все другие реплики того же фрагмента данных должны быть обновлены в той же транзакции. Как правило, синхронная репликация реализуется с помощью триггерных процедур. Создается дополнительная нагрузка при выполнении всех транзакций, в которых обновляются какие-либо реплики. Однако критическим недостатком, полностью исключающим ее применение в рассматриваемых

условиях, является невозможность успешного завершения транзакции, в которой должна обновляться недоступная реплика.

В случае асинхронной репликации обновление одной реплики распространяется на другие спустя некоторое время и не в той же транзакции. При этом имеется задержка (время ожидания), в течение которой отдельные реплики могут быть фактически неидентичными. Как правило, асинхронная репликация реализуется посредством чтения журнала транзакций или очереди тех обновлений, которые подлежат распространению. Ее основное преимущество в том, что дополнительные издержки репликации не связаны с транзакциями обновлений, основной недостаток – данные в различных репликах могут оказаться несовместимыми. Ввиду независимости транзакций в различных репликах возможно возникновение конфликтных ситуаций, которые сложно (иногда даже невозможно) предотвратить и трудно исправить, прежде всего в зависимости от порядка применения обновлений одного и того же фрагмента данных различными репликами. В общем случае задачи устранения конфликтных ситуаций и обеспечения согласованности реплик при асинхронной репликации являются весьма сложными.

Однако особенности репликации фрагментов БД в иерархических ИС являются тем частным случаем, который позволяет избежать таких проблем, прежде всего за счет достаточности master/slave-репликации, базирующейся на предположении, что изменения вносятся лишь в одну главную (master) копию, а остальные зависимые (slave) просто перезаписываются ее содержимым. Этот способ реализует большинство приложений синхронизации, предоставляя пользователю выбор главной копии. При этом копия может представлять собой как полностью БД, так и определенный набор (множество) ее таблиц.

Кроме репликаций с автоматическим распространением обновления одной реплики на остальные, применяется управление копированием без автоматического распространения обновлений. Копии данных создаются и управляются с помощью пакетного или фонового процесса, который отделен во времени от транзакций обновления. Преимуществом такого управления копированием по сравнению с репликацией является большая эффективность ввиду возможности одноразового копирования значительно больших, чем обновляемые в одной транзакции, объемов данных, а недостатком – неидентичность на протяжении большей части времени между моментами синхронизации различных копий данных, что необходимо учитывать в работе с этими данными [1].

Таким образом, приемлемыми способами репликации фрагментов реляционных БД в иерархических ИС с непостоянной связностью узлов следует признать асинхронную master/slave-репли-

кацию и управление копированием. Поскольку первый из них реализует большинство приложений синхронизации БД и требует, как правило, только настройки, а второй подразумевает определенную (в широком диапазоне в зависимости от способа реализации) долю программирования, подробно рассмотрим использование асинхронной master/slave-репликации, в завершение отметив возможности аналогичной реализации управления копированием.

Корректировка исходной структуры БД для слияния внешних и реплицируемых данных

В общем случае реляционным БД иерархических ИС присущи черты как оперативных БД, в основном содержащих текущие данные, так и БД поддержки принятия решений и хранилищ данных, содержащих архивы исторически накопленных данных. Для БД различных уровней иерархии ИС важно определение уровня детализации БД, то есть самого низкого уровня обобщения данных, предназначенных для хранения в БД. При информационном взаимодействии узлов смежных уровней иерархии ИС уровень детализации БД не должен понижаться с повышением уровня иерархии. В такой БД содержатся структуры, отражающие иерархию ИС и объектов (сущностей модели данных), информация о которых хранится в БД, в привязке к узлам ИС, формирующим эту информацию. При выполнении процедуры консолидации любая неявная связь между данными из разных источников преобразуется в явную связь, например, путем введения явных значений унифицированных идентификаторов.

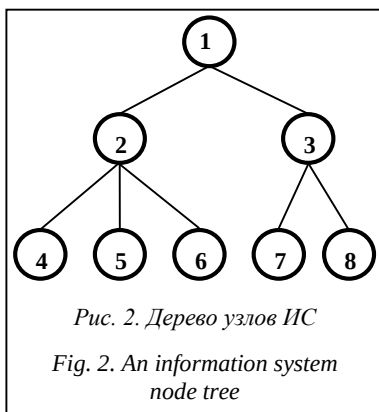
Информация из внешних для ИС источников используется только при отсутствии связи с узлами ИС, ответственными за ее формирование. Обязательным условием отсутствия неявных связей, гарантирующим консолидацию данных, является идентичность первичных или альтернативных ключей. При полноте такой информации, относящейся к объекту или узлу ИС, соответствующая информация в БД полностью замещается. При ее частичности обязательно должен передаваться перечень значений ключей удаленных данных (совпадающие значения ключей используются для обновления данных, а несовпадающие – для вставки).

С учетом изложенного без существенного уменьшения общности с одновременным сохранением наглядности можно ограничить рассмотрение обобщенной (по узлам ИС) исходной структуры БД следующими таблицами: **HN** – иерархии узлов ИС; **HO** – иерархии объектов, описываемых в ИС; **MI** – основной информации об объекте; **DI** – детализированной информации об объекте. Особенности структур этих таблиц находятся вне области нашего рассмотрения, за следующим исключе-

нием. Внешними ссылками в таблицах моделируются обязательные связи типа «многие к одному» от каждой последующей таблицы данного списка к предыдущей и необязательные связи иерархии в первых двух таблицах списка (рис. 1). Иерархия объектов отражает их структурную вложенность, а ссылка от объекта к узлу указывает источник первичной информации по этому объекту в ИС. Во всех таблицах присутствует флаг признака получения информации из внешнего для ИС источника. Естественно, что в реальных БД содержится множество таблиц, подобных двум последним как по параллельности, так и по последовательности связей, в том числе и с обобщенными данными, но для рассмотрения сущности предлагаемого подхода этих таблиц достаточно.

Основными причинами корректировки предложенной исходной структуры БД являются ограничения асинхронной master/slave-репликации и обеспечение совместимости реплицируемой информации с информацией из внешних источников. Прежде всего это единственность (невозможность множественности) master-таблицы при возможной множественности slave-таблиц и ограниченность детализации репликации отдельной таблицей, а не выборкой или проекцией из таблицы. Поэтому отдельные таблицы должны выделяться для данных, относящихся как к master-, так и к slave-информации хотя бы в одном репликационном наборе (множестве таблиц). Таким образом, в преобразованной структуре БД узла ИС вместо каждой таблицы исходной структуры БД должны содержаться несколько идентичных по структуре таблиц в зависимости от связности с другими узлами ИС, а также для информации из внешних источников.

Для сохранения наглядности без существенного уменьшения общности рассмотрим структуру ИС деревом, изображенным на рисунке 2, в вершинах которого указаны условные номера узлов ИС.



В качестве основного преобразования исходной структуры БД предлагается создавать для каждой таблицы, входящей в какой-либо репликационный набор, наследуемые от нее таблицы, по одной на каждый узел репликации, а также для внешнего источника информации. Одиночное наследование таблиц описывается в стандартах SQL и поддерживается практически всеми современными реляционными СУБД [1]. Данные предлагается хранить в дочерних таблицах, а СПО обращаться к данным в родительской таблице (без опции ONLY, то есть с обращением к потомкам). При этом происхождение данных однозначно определяется дочерней таблицей их хранения, в которой и производится изменения данных, репликационно или от внешнего источника информации. Освободившийся в структуре таблиц флаг признака внешнего источника информации семантически переопределяется во флаг актуальности текущей записи, управляемый унифицированными триггерными процедурами для дочерних таблиц. В простейшем случае любые изменения устанавливают данный флаг в текущей записи и снимают его во всех других записях с таким же значением ключа. Фильтрация по значению флага при обращении к данным в родительской таблице позволяет практически без корректировки использовать СПО, предназначенное для работы с исходной структурой БД. Введение в структуру таблиц аналогично управляемых дополнительных флагов позволит различной фильтрацией данных по их значениям одновременно имитировать различные, в том числе существенно более сложные, чем только по флагу актуальности, стратегии слияния данных для различных задач СПО. А введение управляемых пользователем флагов предоставляет возможность произвольного (по усмотрению пользователя) совмещения данных.

Для обобщенной исходной структуры БД преобразованная таким образом структура БД (рис. 2) на узле с условным номером 3 промежуточного уровня иерархии, связанном с узлом вышестоящего уровня с условным номером 1 и двумя узлами нижестоящего уровня с условными номерами 7 и 8, будет иметь вид, представленный на рисунке 3. В дополнение к внешним ссылкам пунктирными стрелками на рисунке 3 показано наследование таблиц.

В преобразованной структуре БД имена дочерних таблиц состоят из исходных имен родительских таблиц, дополненных символом V для таблиц с информацией из внешних источников или условными номерами узлов для таблиц с реплицируемыми данными. Подобное преобразование структуры БД может быть легко автоматизировано и не представляет сложности. Единственное дублирование данных происходит при заполнении таблиц информацией, реплицируемой с данного узла, поскольку она является вторичной от информации из внешних источников или реплицируемой от смеж-

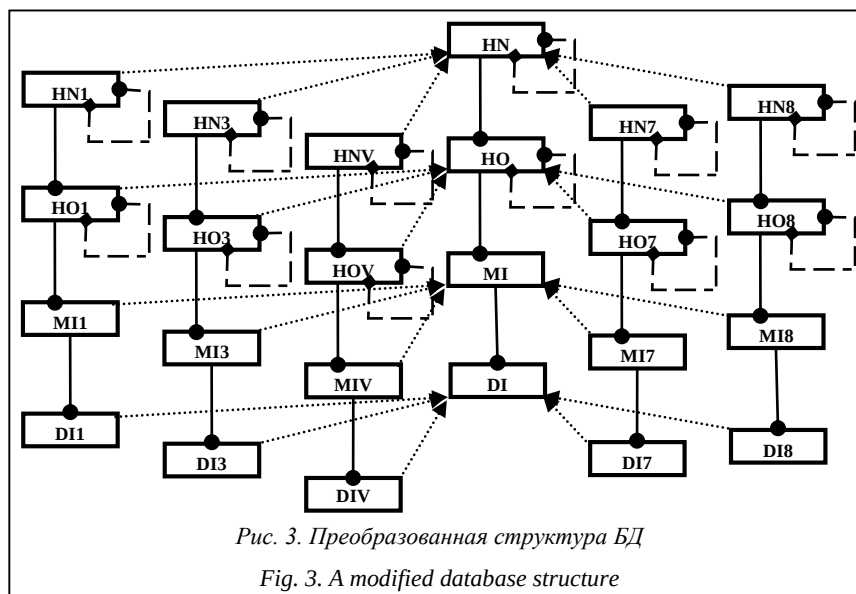


Рис. 3. Преобразованная структура БД

Fig. 3. A modified database structure

ных узлов. Все изменения в этих таблицах инициализируются триггерами других таблиц, дочерних от соответствующей родительской таблицы, и выполняются унифицированными процедурами, реализующими с применением флагов выбранную стратегию слияния и при необходимости фильтрации данных. Типичным условием фильтрации данных для репликации является специализация информации. В рассматриваемом примере для репликации нижестоящим по иерархии узлам ИС данные об иерархии узлов ИС могут быть ограничены одним вышестоящим и всеми подчиненными нижестоящими по иерархии узлами.

При предположении, что в рассматриваемой иерархической ИС информация передается только между смежными по иерархии узлами, причем информация об иерархии узлов – сверху вниз, а остальная информация – снизу вверх по иерархии, таблицы **HO3**, **MI3** и **DI3** узла с условным номером 3 будут являться master-копиями в репликации на узел с условным номером 1. Таблица **HN3** (при отсутствии фильтрации данных) или таблицы **HN7** и **HN8** (при наличии фильтрации данных) будут являться master-копиями в репликации на узлы с условными номерами 7 и 8. Таблица **HN1** будет являться slave-копией в репликации от узла с условным номером 1. Таблицы **HO7**, **MI7** и **DI7** будут являться slave-копиями в репликации от узла с условным номером 7, а таблицы **HO8**, **MI8** и **DI8** – от узла с условным номером 8.

Таблицы **HO1**, **MI1**, **DI1** и таблица **HN3** или таблицы **HN7** и **HN8** в зависимости от фильтрации данных на узле с условным номером 3 не используются и могут отсутствовать. Аналогично при описанном подходе к преобразованию структуры БД неиспользуемыми будут таблицы в БД и на других узлах ИС. Однако неиспользуемые таблицы не являются существенным отягощением БД, в связи с чем предлагается унифицированная для всех уз-

лов ИС структура БД, являющаяся объединением структур БД всех узлов ИС и позволяющая в максимально возможной степени унифицировать СПО решения функциональных задач на различных узлах ИС, возложив функции репликации и консолидации данных на систему асинхронной master/slave-репликации и хранимые (триггерные и вызываемые из триггеров) процедуры СУБД. Дополнительным преимуществом такого подхода является возможность изменения уровней детализации БД различных узлов ИС перенастройкой

множеств реплицируемых таблиц и их подписки.

Пример реализации описанного подхода с использованием сертифицированных ПС

Рассмотрим пример реализации ключевых аспектов описанного подхода к репликации реляционных БД в иерархических ИС с непостоянной связностью узлов средствами конкретных СУБД и системы асинхронной master/slave-репликации управляемых ею БД. Учитывая тенденцию к увеличению независимости от проприетарных импортных программных продуктов, а также требования использования в разрабатываемых ИС отечественных программных средств (ПС), во многих случаях имеющих определенные сертификаты безопасности и разрешения использования, выберем объектно-реляционную СУБД PostgreSQL с системой асинхронной master/slave-репликации Slony-I [2, 3]. Такой выбор обусловлен их соответствием в основном современным стандартам в области реляционных СУБД, подтвержденной широким использованием эффективностью [4], а также свободным распространением в исходных кодах и вхождением в состав многих отечественных сертифицированных ПС, таких как операционные системы MCBC и Astra Linux [5, 6]. Пример будет базироваться на совместимых между собой версиях PostgreSQL и Slony-I, входящих в состав последних версий этих ПС.

Без потери общности полагаем, что в примере все таблицы создаются в одной схеме **sch** в одноименной БД **dbexam** для всех узлов ИС с условными номерами **unb** от 1 до 8, имеющими адрес '333.22.unb.1'. Для реплицирования везде используются порт по умолчанию и пользователь **postgres** с паролем **pres**. Имена таблиц в сопровождающем тексте приведены прописными буквами исключительно для наглядности.

Первоначально для создания таблиц по наследованию от заданных в PostgreSQL для каждой из таких таблиц выполняется команда типа

```
CREATE TABLE sch.hn1 ([ ]) INHERITS (sch.hn);
```

В этих командах вместо квадратных скобок перечисляются не наследуемые по умолчанию, но необходимые в наследуемых таблицах ограничения, такие как первичные и внешние ключи, уникальности, а также значения по умолчанию. Например, для поля **f1** первичного ключа, поля **f2** альтернативного ключа и целочисленного поля **f3** со значением по умолчанию, равным условному номеру узла ИС, для создания таблицы **HN1** на узле 1 выполняется команда

```
CREATE TABLE sch.hn1 (
  f3 integer DEFAULT 1, PRIMARY KEY (f1),
  UNIQUE (f2)
) INHERITS (sch.hn);
```

Соответственно на узле 3 выполняется команда

```
CREATE TABLE sch.hn1 (
  f3 integer DEFAULT 3, PRIMARY KEY (f1),
  UNIQUE (f2)
) INHERITS (sch.hn);
```

Отметим, что внешние ключи дочерних таблиц должны ссылаться не на родительские, а на соответствующие дочерние таблицы. Так, при существовании в таблице **HO** наряду с полем **f1** первичного ключа поля **f4** внешней ссылки на таблицу **HN**

```
FOREIGN KEY (f4) REFERENCES sch.hn(f1)
```

для создания таблицы **HO1** на любом узле выполняется команда

```
CREATE TABLE sch.ho1 (
  PRIMARY KEY (f1), FOREIGN KEY (f4)
  REFERENCES sch.hn1(f1)
) INHERITS (sch.ho);
```

Аналогичные действия выполняются для всех остальных таблиц. В результате в схеме **sch** БД **dbexam** на каждом из 8 узлов ИС наряду с 4 исходными таблицами **HN**, **HO**, **MI** и **DI** будут созданы 36 таблиц **HN(1–8, V)**, **HO(1–8, V)**, **MI(1–8, V)** и **DI(1–8, V)**, где **MI(3–5, V)**, например, обозначает список таблиц **MI3**, **MI4**, **MI5** и **MI4**.

Кроме того, должна быть разработана унифицированная триггерная процедура, управляющая на каждом узле корректировкой данных в таблицах, являющихся master-копиями в репликации на внешние узлы при изменении данных в таблицах, являющихся slave-копиями в репликации от внешних узлов и таблицах с информацией из внешних источников данного узла. Данная процедура должна активизироваться соответствующими триггерами и реализовывать выбранную стратегию консолидации и фильтрации для внешнего представления данных, использующую введенные флаги. Учитывая интенсивность и условия выполнения, особое внимание должно уделяться эффективности выполняемых в ней запросов, в которых данные фильтруются по значениям ключей и флагов. В этом случае при больших объемах данных весьма эффективны многоколоночные индексы, включающие поля флагов после ключевых полей [7].

Эта процедура является основной специализированной разработкой. Все остальные приведенные действия легко автоматизируются, в связи с чем создание соответствующего диалогового приложения, в котором описываются узлы ИС и перечисляются используемые в репликациях таблицы, не представляет проблемы. Естественно, что добавление в ИС новых узлов потребует добавления соответствующих объектов в БД, причем каждого узла, поскольку отсутствуют репликации изменений, внесенных DDL-командами. Другими наиболее существенными ограничениями последних версий системы Slony-I, которые необходимо учитывать, являются отсутствие репликаций изменений для больших объектов (BLOB) и сделанные командой TRUNCATE, а также изменения пользователей и ролей.

По окончании преобразования БД задается имя репликационного кластера, например **slony_example**, с которым связываются все дальнейшие действия по его настройке, выполняемые с помощью утилиты командной строки Slonik конфигурации системы Slony-I. Разумеется, в каждом узле ИС, кроме СУБД PostgreSQL, должна быть установлена система Slony-I асинхронной master/slave-репликации, а в случае отсутствия в каждой БД должен быть установлен необходимый этой системе процедурный язык PL/pgSQL.

Настройка репликационного кластера **slony_example** заключается в передаче утилите Slonik скриптов на ее командном языке (из текстового файла или стандартного входа), начинающихся с преамбульной команды указания кластера

```
CLUSTER NAME = slony_example;
```

В начале настройки все узлы репликационного кластера, в данном случае 8 узлов ИС с доступом к их БД, описываются также преамбульной командой описания этих узлов. Например, узел 3 описывается командой

```
NODE 1 ADMIN CONNINFO = 'dbname=dbexam
host=333.22.3.1 user=postgres password=pres';
```

Остальные команды в командном языке утилиты Slonik относятся к конфигурационным или командам действий. Прежде всего командой INIT CLUSTER (id=1, comment = 'Master Node 1'); создается кластер на master-узле. При этом в БД создается схема **_slony_example**, содержащая все необходимые репликационной системе объекты БД. Затем в созданный кластер добавляются slave-узлы. Например, узел 3 добавляется командой

```
STORE NODE (id=3, comment = 'Slave node 3',
event node=1);
```

После этого устанавливаются пути доступа от каждого узла ко всем остальным узлам, которые могут использоваться. Так, для пары узлов 1 и 3 должна быть выполнена пара команд:

```
STORE PATH (server=1, client= 3, conninfo=
'dbname=dbexam host=333.22.1.1
user=postgres password=pres';
STORE PATH (server=3, client= 1, conninfo=
'dbname=dbexam host=333.22.3.1
user=postgres password=pres';
```

В командах термины `server` и `client` относятся не к роли узлов в кластерной конфигурации, а к источникам и получателям сообщений или данных.

Далее создаются репликационные наборы таблиц, на которые осуществляется подписка узлов. В данном случае следует создать по два набора для узлов 2 и 3 как промежуточных в иерархии ИС и по одному набору на остальные узлы, так как на каждом узле имеются таблицы, на репликацию которых подписан хотя бы один узел, и подписки в одном направлении иерархии идентичны. Репликационные наборы для узлов 1, 3 и 7 создаются командами

```
CREATE SET (id=1, origin=1);
CREATE SET (id=4, origin=3);
CREATE SET (id=5, origin=3);
CREATE SET (id=9, origin=7);
```

Каждый репликационный набор описывается перечнем входящих в него таблиц (и, возможно, последовательностей) путем добавления их к набору. Таблицам присваиваются уникальные идентификаторы среди всех наборов кластера. Порядок включения таблиц в набор определяет порядок их обработки. В рассматриваемом примере в перечисленные наборы будут добавлены таблицы следующими командами:

```
SET ADD TABLE (set id=1, id=1,
fully qualified name='sch.hn1');
SET ADD TABLE (set id=4, id=6,
fully qualified name='sch.hn3');
SET ADD TABLE (set id=5, id=7,
fully qualified name='sch.ho3');
SET ADD TABLE (set id=5, id=8,
fully qualified name='sch.mi3');
SET ADD TABLE (set id=5, id=9,
fully qualified name='sch.di3');
SET ADD TABLE (set id=9, id=19,
fully qualified name='sch.ho7');
SET ADD TABLE (set id=9, id=20,
fully qualified name='sch.mi7');
SET ADD TABLE (set id=9, id=21,
fully qualified name='sch.di7');
```

Наконец, выполняется подписка на созданные наборы таблиц. На представленные наборы в соответствии с логикой примера подписка выполняется следующими командами:

```
SUBSCRIBE SET (id=1, provider=1, receiver=3,
forward=no);
SUBSCRIBE SET (id=4, provider=3, receiver=7,
forward=no);
SUBSCRIBE SET (id=5, provider=3, receiver=1,
forward=no);
SUBSCRIBE SET (id=9, provider=7, receiver=3,
forward=no);
```

На этом настройка репликационного кластера завершена. Для его функционирования на всех узлах кластера должен выполняться процесс `slon`, которому в качестве параметра переданы имя кластера и информация соединения с БД. В принципе, процесс `slon` и БД, к которой он подсоединяется, могут располагаться на различных узлах, но при непостоянной связности узлов это лишь создает дополнительные проблемы. Предлагается запускать этот процесс на каждом узле ИС как сервис после запуска сервиса СУБД и останавливать, соответственно, перед остановкой сервиса СУБД.

Основными условиями нормального функционирования системы Slony-I являются доступность всех узлов, синхронизация их часов, предпочтительно с использованием таких стабильных часовых поясов, как UTC или GMT, и использование общей кодировки БД. Проведенные над созданным описанным способом репликационным кластером со стабильным часовым поясом и общей кодировкой БД эксперименты показали корректное восстановление репликации после многочасовых прерываний связи между узлами.

Однако если параметры рассинхронизации часов или непостоянства связности узлов ИС не позволяют настроить систему Slony-I асинхронной master/slave-репликации многочисленными конфигурационными командами утилиты Slonik таким образом, чтобы обеспечивались требуемые показатели функционирования ИС, то необходимо реализовывать собственный механизм управляемого копирования, позволяющий разрешать такие проблемы. В последних версиях СУБД PostgreSQL в качестве основы такого механизма к модулю `dblink` поддержки сессионного соединения с другими БД добавился модуль `postgres_fdw` обертки внешних данных, функциональность которого в основном перекрывает функциональность модуля `dblink`, предоставляя более стандартный синтаксис доступа к удаленным данным и обеспечивая во многих случаях более высокую производительность. Принципиальным недостатком применения этих модулей по сравнению с системой репликации является смещение акцента выполняемых действий с настройки на программирование. При этом необходимо решать не только задачи выявления обновлений данных на сервере и их актуализации на клиенте, но и такие скрытые системой Slony-I задачи, как управление поведением исходных триггеров и правил реплицируемых таблиц и управление блокировками. Несомненным преимуществом управляемого копирования является возможность обращения к отдельным записям таблиц и отказа от описанного в статье преобразования структуры БД. Тем не менее, для разделения функций реплицирования и консолидирования данных предлагается использовать такое преобразование структуры БД, в том числе и при управляемом копировании.

Определенный интерес может представлять совместное использование в ИС репликации и управляемого копирования на непересекающихся фрагментах БД, позволяющее в наибольшей мере использовать их преимущества. Например, представляется целесообразным реплицировать интенсивно изменяющиеся небольшие по объему данные и управляемо копировать без какого-либо дублирования консервативные специализированные по доступу большие объемы информации.

Наконец, нельзя не упомянуть о докладе на последней Общероссийской конференции по PostgreSQL в Москве [4] Симона Риггза (Simon Riggs), од-

ного из создателей разрабатываемой в открытых кодах новейшей репликационной технологии BDR (BiDirectional Replication – двунаправленная репликация) для ядра PostgreSQL. В докладе утверждалась работоспособность технологии BDR с ядром 9.4 PostgreSQL и описывалось ее становление в качестве основной репликационной технологии для PostgreSQL с последующим отказом от дальнейшего развития системы Slony-I. Однако в качестве полноценной multi-master-репликации эта технология планируется для еще не выпущенного (на момент написания статьи) ядра 9.6 PostgreSQL. Поскольку существует естественное отставание версий ядра PostgreSQL в сертифицированных ПС, применение технологии BDR (с возможной адаптацией под нее предлагаемого в статье подхода) к репликации реляционных БД в иерархических ИС с непостоянной связностью узлов может являться лишь предметом дальнейшего рассмотрения.

На основании изложенного можно сделать вывод о том, что предлагаемый подход к репликации реляционных БД в иерархических ИС с непостоянной связностью узлов позволяет существенно отделить проблему поддержания целостности и актуальности данных в таких системах от решаемых на этих данных функциональных задач.

Введение дополнительной избыточности в структуру БД за счет наследования специализированных под конкретный источник данных таблиц повышает возможности применения в таких системах асинхронной master/slave-репликации, существенно сокращая объем разрабатываемого СПО, а унификация структуры БД для различных узлов ИС предоставляет дополнительную возможность унификации СПО.

Литература

1. Дейт К.Дж. Введение в системы баз данных; [пер. с англ.]. М.: Вильямс, 2005. 8-е изд. 1328 с.
2. PostgreSQL. Documentation. URL: <http://www.postgresql.org/docs> (дата обращения: 15.06.2015).
3. Slony-I enterprise-level replication system. Documentation. URL: <http://slony.info/documentation> (дата обращения: 15.06.2015).
4. Докл. Рос. конф. разработчиков и пользователей PostgreSQL междунар. уровня (PgConf.Russia). 2015. URL: <http://pgconf.ru/2015> (дата обращения: 17.02.2015).
5. Вопросы по MCBC. URL: <http://www.vniins.ru/index.php?s=вопросы%20по%20mcbc&lang=Рус> (дата обращения: 15.06.2015).
6. Astra Linux – операционная система специального назначения на базе ядра Linux. URL: <http://www.astra-linux.com> (дата обращения: 15.06.2015).
7. Сорокин В.Е. Метод искусственного соответствия SQL-запросов индексам реляционных баз данных // Программные продукты и системы. 2013. № 2. С. 47–54.

DOI: 10.15827/0236-235X.112.203-209

Received 01.07.15

DATA REPLICATION IN HIERARCHICAL INFORMATION SYSTEMS WITH UNSTABLE NODE CONNECTIVITY

Sorokin V.E., Ph.D. (Engineering), Leading Research, sorokinve@yandex.ru

(R&D Institute «Tsentrprogrammssystem», 50 let Oktyabrya Ave. 3a, Tver, 170024, Russian Federation)

Abstract. A particular feature of many hierarchical information systems (IS) with relational databases (DB) in their nodes as the core is impermanent communication between the nodes. It prevents classical use of a centralized or distributed database and assigns additional special message exchange and data consolidation functions including external sources to special software, which solves system functional tasks.

Replication of required fragments of other IS nodes DB in a local DB consolidates with information from external sources. It allows to simplify and increase harmonization of various IS nodes software. The article proposes an approach to original DB structure transformation based on creating separate for different data sources inherited from source tables. The approach is adapted to asynchronous “master/slave” replication of hierarchically related DB. The paper shows the possibility of using flags and procedures managed by triggers to implement various strategies for data consolidation, as well as a simple adaptation of the original software to solve IS functional problems on the converted DB. There is an example of the described approach implementation for widespread and used in many domestic certified software tools of an object-relational database management system PostgreSQL with an asynchronous “master/slave” replication system Slony-I. There is a possibility to overcome limitations of an asynchronous “master/slave” replication system using dblink support modules of session connection to other DB and postgres_fdw foreign-data wrapper, as well as replication system joint using with these modules on the converted DB.

Keywords: hierarchical information system, relational database, asynchronous replication, inheritance tables, external data wrapper.

References

1. Date C.J. *An Introduction to Database Systems*. 8th ed. Person/Addison Wesley Publ., 2004 (Russ. ed.: Moscow, Vilyams Publ., 8th. ed., 2005, 1328 p.).
2. PostgreSQL. Documentation. Available at: <http://www.postgresql.org/docs> (accessed June 15, 2015).
3. Slony-I enterprise-level replication system. Documentation. Available at: <http://slony.info/documentation> (accessed June 15, 2015).
4. PGCONF.RUSSIA 2015. *Russian PostgreSQL developers and users conf*. Moscow, 2015. Available at: <http://pgconf.ru> (accessed February 17, 2015).
5. *Voprosi po MCBC* [VNIINS. MCBC Questions]. Available at: http://www.vniins.ru/menu/question_mcbc_rus (accessed June 15, 2015).
6. *Astra Linux – operatsionnaya sistema spetsialnogo naznacheniya na baze yadra Linux* [Astra Linux.Linux-based Special-purpose Operation System]. Available at: <http://www.astra-linux.com> (accessed June 15, 2015).
7. Sorokin V.E. A method of artificial matching of sql query to relational databases indexes. *Programmnye produkty i sistemy* [Software and Systems]. 2013, no. 2, pp. 47–54 (in Russ.).

004.043

Дата подачи статьи: 30.09.15

DOI: 10.15827/0236-235X.112.210-218

**МОДЕЛИ ПРЕДСТАВЛЕНИЯ ДАННЫХ
СЛОЖНЫХ ПРОИЗВОДСТВЕННЫХ ПРОЕКТОВ
В АВТОМАТИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ
ПРОМЫШЛЕННЫХ ПРЕДПРИЯТИЙ**

(Работа выполнена при финансовой поддержке РФФИ, проект № 15-07-02935 А)

М.И. Дли, д.т.н., профессор, midli@mail.ru;

О.В. Стоянова, к.э.н., доцент, ovstoyanova@list.ru

(Смоленский филиал Национального исследовательского университета МЭИ,
Энергетический проезд, 1, г. Смоленск, 214013, Россия);

А.Ю. Белозерский, д.э.н., профессор

(Российский химико-технологический университет им. Д.И. Менделеева,
Миусская пл., 9, г. Москва, 125047, Россия)

В процессе управления сложными производственными проектами на промышленных предприятиях используется множество разнообразных информационных систем, каждая из которых решает отдельные задачи обработки проектных данных. Так, для проектов по организации производства новых образцов продукции сведения об изделии распределены между САПР и АСУ производством, информация о необходимых производственных ресурсах – между АСУ производством, АСУ предприятием, АСУ проектами и т.д. Обмен проектными данными между указанными системами осложняется отличиями используемых ими программно-аппаратных платформ, форматов данных, моделей представления этих данных.

В статье рассмотрены два вида моделей представления данных сложных производственных проектов: объектная и реляционная. Модель первого вида используется в качестве основы обработки данных с помощью объектных языков программирования, реляционная модель необходима для организации хранения данных в БД промышленных предприятий. Данный выбор обусловлен современными тенденциями развития информационных технологий в сфере автоматизации промышленных предприятий: использование объектного программирования для реализации приложений по обработке данных и доминирование реляционных СУБД.

Использование различных моделей данных в отдельных компонентах информационной системы требует разработки решений по обеспечению преобразования данных из одного представления в другое. Указанная задача не имеет на сегодняшний день универсального решения. В статье рассмотрен пример такого преобразования для объектной модели, реализованной в Java, и реляционной модели для СУБД Oracle.

Ключевые слова: производственный проект, модель проектных данных, управление проектами, автоматизированная информационная система.

Комплексная автоматизация деятельности промышленного предприятия имеет свои особенности и предполагает наличие систем следующих типов: АСУ, САПР, автоматизированные системы научных исследований (АСНИ), информационно-аналитические системы (ИАС) и информационные системы поддержки принятия решений (ИСППР).

В свою очередь, в классе АСУ принято выделять АСУ технологическими процессами (АСУ ТП) (SCADA), АСУ производством (АСУ П) (MES), АСУ предприятиями (АСУП) (ERP).

SCADA-системы (Supervisory Control and Data Acquisition, диспетчерское управление и сбор данных) предназначены для управления удаленными техническими объектами в режиме реального времени. Данные SCADA-систем собираются в оперативных БД предприятия и анализируются в процессе управления производством, в том числе с использованием MES-систем (Manufacturing Execution System, система управления производственными процессами).

MES-система позволяет решать задачи координации, синхронизации, анализа и оптимизации процессов выпуска продукции [1, 2]. MES-системы получают задания на планирование процессов от

ERP-систем (Enterprise Resource Planning, системы управления ресурсами предприятия).

ERP-системы предназначены для управления всеми видами ресурсов предприятия. Наиболее распространенными среди таких систем являются BAAN (компания BAAN), R/3 (SAP), Oracle Applications (Oracle), SAS System (SAS Institute), Галактика (корпорация «Галактика»), 1С: Управление производственным предприятием («1С»), М-3 (К-Т-С). Функции ERP-систем могут быть расширены за счет интеграции с другими системами управления предприятием, например, SCM – планирование цепочек поставок, SSM – продажи и управление сервисом.

Распределение основных функций управления проектами между указанными системами показано в таблице 1.

На основании таблицы можно сделать следующие выводы:

- многие задачи управления производственными проектами требуют использования возможностей нескольких систем;
- некоторые задачи решаются с применением данных систем лишь частично, например, планирование работы может быть выполнено с использова-

нием MES-системы лишь для производственных работ;

– ряд задач проектного управления невозможно решить посредством перечисленных систем, они требуют использования ИСУП, ИАС и ИСППР.

Таблица 1

**Реализация функций управления проектами
в автоматизированных системах предприятия**

Table 1

**Projects management functions implementation
in enterprise automated systems**

Функция	Класс систем, подсистемы
Планирование работ проекта	MES, Оперативное/Детальное планирование производства (ODS)
Планирование и контроль материальных ресурсов проекта	MES, Контроль состояния и распределение производственных ресурсов (RAS) MES, Управление производственными фондами (техобслуживание) (MM)
Распределение работ между исполнителями	MES, Управление персоналом (LM) ERP, Управление человеческими ресурсами (HR)
Планирование организационной структуры управления проектом	ERP, Управление человеческими ресурсами (HR)
Планирование и контроль исполнения бюджета	ERP, Управление финансами (FM)
Контроль качества продукции и процессов	MES, Управление качеством продукции (QM)
Анализ рисков	–
Анализ сценариев проекта	–

ИСУП (PMS – Project Management System) предназначены для планирования и мониторинга проектов различных типов.

Существует множество программных средств PMS, ориентированных как на решение отдельных задач проектного управления, таких как планирование бюджета, контроль выполнения плана и т.д., так и на весь жизненный цикл проекта. Стабильно на российском рынке присутствуют около 10 систем. Среди лидеров такие общепризнанные решения, как Microsoft Enterprise Project Management Solution (EPM), Oracle Primavera, Deltek Open Plan. Есть и отечественные разработки, в том числе Spider Project от компании «Технологии управления Спайдер», Project Kaiser компании Triniforce. Сравнительный анализ данных систем приведен в работе [3].

Для решения аналитических задач проектного управления возможно использование систем OLAP (Online Analytical Processing, аналитическая обработка в реальном времени) и BI (Business Intelligence, бизнес-аналитика), однако подобные си-

стемы пока не получили широкого распространения на отечественных промышленных предприятиях, хотя примеры их использования существуют [4].

В последнее время большое внимание уделяется интеграции систем АСУ и САПР. Это связано с тем, что в числе наиболее важных исходных данных для MES-систем технологические процессы производства [5, 6].

В соответствии с ГОСТ 23501.101-87 «Системы автоматизированного проектирования. Основные положения», САПР – это организационно-техническая система, входящая в структуру проектной организации и осуществляющая проектирование при помощи комплекса средств автоматизированного проектирования.

В зависимости от разных признаков классификации можно выделить несколько классов систем САПР [7]. Наиболее часто применяемый признак целевого назначения данных систем разделяет их по аспектам проектирования на CAD (двумерное и/или трехмерное геометрическое проектирование, создание конструкторской и/или технологической документации), CAE (системы автоматизации инженерных расчетов, анализа, симуляции физических процессов, динамического моделирования, проверки и оптимизации изделий), CAM (системы автоматизации программирования и управления оборудованием с ЧПУ или ГАПС) и CAPP (системы автоматизации планирования технологических процессов).

Обеспечить преемственность конструкторских и технологических решений в САПР позволяет использование PDM-систем (Product Data Management, управление данными о продукции). Это системы управления информацией об изделии, включая инженерные и технические данные.

Примерами российских и западных PDM-систем являются IC: Интегратор, IC: PDM Управление инженерными данными, Lotsia PDM Plus, T-FLEX DOCs, TechnologiCS, Windchill, SmartTeam, IDPM CADISON PDM, SolidWorks Enterprise PDM.

В последнее время в сфере промышленной автоматизации наблюдается переход на интегрированные платформы PLM (Product Lifecycle Management, управление жизненным циклом продукции). При функционировании PLM опирается на использование совокупности рассмотренных выше информационных систем. Лидерами на рынке PLM являются Teamcenter (Siemens PLM Software), Agile PLM (Oracle), решения от Dassault Systemes. Данные решения являются дорогостоящими, поэтому область их внедрения ограничивается на сегодняшний день крупными предприятиями и концернами.

На основании представленного описания можно выделить следующие особенности автома-

тизации деятельности промышленных предприятий, обусловленные использованием множества разнородных информационных систем:

- сложность информационной инфраструктуры, обусловленная использованием различных программно-аппаратных платформ;
- постоянное развитие информационных систем в направлении расширения функциональных возможностей и увеличения производительности;
- дублирование ряда функций или наличие нескольких их реализаций;
- избыточность хранимой информации в связи с ее дублированием в отдельных системах;
- использование различных представлений одних и тех же данных и необходимость их согласованного изменения.

Решения, предлагаемые авторами, разработаны с учетом перечисленных особенностей и направлены на снижение избыточности хранимой информации и обеспечение согласованного изменения представленных в моделях данных.

Реляционная модель данных производственного проекта

Для управления сложным производственным проектом необходимо иметь полный объем плановой и оперативной информации, характеризующей данный проект, а именно сведения о работах, про-

цессах, целях, событиях, мероприятиях, а также о результатах, получаемых на каждой стадии проекта [8].

Данная информация хранится в БД автоматизированной информационной системы, поэтому при проектировании ее структуры необходимо принимать во внимание тип СУБД, используемой на предприятии. В зависимости от используемой модели данных выделяют пять видов СУБД: иерархические, сетевые, реляционные, объектные и объектно-реляционные. Несмотря на широкие возможности СУБД последних двух типов, самыми используемыми на отечественных предприятиях являются реляционные СУБД, такие как Microsoft SQL Server, Oracle Database, Teradata Database, DB2 и др.

Разработка структуры реляционной БД включает построение логической и физической моделей. Логическая модель отражает понятия предметной области и их связи между собой. В ходе проведенного анализа было выделено 11 основных сущностей, описание которых приведено в таблице 2, а графическое представление дано на рисунке 1.

Как видно из рисунка 1, между многими сущностями существует связь «многие-ко-многим», то есть одному элементу первой сущности соответствует множество элементов второй и наоборот. Данный вид связи является естественным для производственных проектов, однако ее нельзя отра-

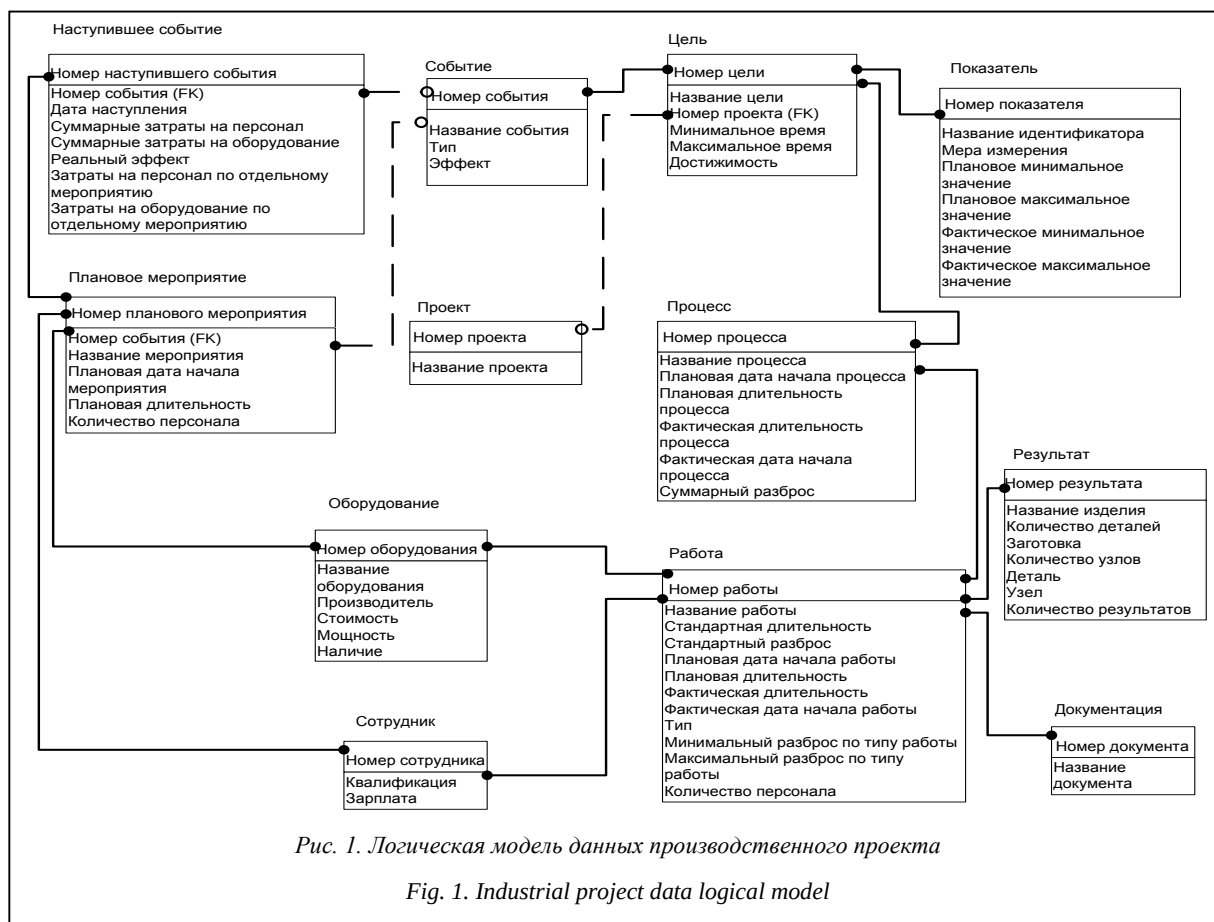


Рис. 1. Логическая модель данных производственного проекта

Fig. 1. Industrial project data logical model

зить в БД. В связи с этим физическая модель реляционной БД (рис. 2) отличается от логической.

Таблица 2

Сущности логической модели данных

Table 2

Data logical model entities

Название	Описание
Проект	Информация обо всех производственных проектах
Цель	Цели, которых необходимо достигнуть для реализации производственных проектов
Показатель	Перечень показателей, по которым оценивается достижимость целей
Событие	Сущность, включающая в себя события, которые могут повлиять на достижение цели
Наступившее событие	Информация о реально наступивших событиях, степени влияния события на цель и затраты, связанные с ним
Мероприятие	План мероприятий, которые могут быть проведены в случае наступления какого-то конкретного события
Процесс	Перечень процессов, которые необходимо выполнить для достижения целей
Работа	Совокупность работ, входящих в конкретные процессы производственного проекта
Результат	Вид и перечень результатов, которые могут быть получены при выполнении разных работ
Оборудование	Перечень оборудования, задействованного в конкретной работе
Сотрудник	Перечень сотрудников, задействованных в конкретной работе
Документация	Сопроводительная документация по работам производственного проекта

Как видно из рисунка 2, для организации связи «многие-ко-многим» дополнительно создаются связующие таблицы, такие как REAL_EVENT_MER, AIM_EVENT_MEROPRIYT, AIM_PROCESS, PROC_WORK, WORK_PERSON, WORK_EQUIPMENT, PERSON_MEROPR, EQUIPMENT_MEROPR. Кроме этого, создаются еще две справочные таблицы, отражающие тип события и тип работы: EVENT_TYPE и WORK_TYPE.

Как видно из рисунка 2, представление данных о проекте в реляционном виде приводит к необходимости создания большого количества таблиц, многие из которых предназначены исключительно для физической реализации связей типа «многие-ко-многим».

Объектная модель данных производственного проекта

Как уже отмечалось, ряд задач управления сложными производственными проектами невозможно решить с использованием информационных систем, применяемых на отечественных промышленных предприятиях. Их решение требует разработки дополнительного ПО, реализующего необходимые алгоритмы. На сегодняшний день ведущая

роль в сфере разработки прикладных программ принадлежит объектным языкам программирования, к числу которых относится Java.

Объектная модель данных производственного процесса, ориентированная на создание программных средств на языке Java, приведена на рисунке 3. Данная модель представляет собой диаграмму классов в соответствии с методологией UML.

На диаграмме видим, что между классами существуют четыре вида отношений.

Отношения обобщения (заканчиваются стрелкой) отражают факт наследования классов Real_event, Indicator_Value и Result от родительских и их расширения от него. Расширение реализуется через ключевое слово «extends». Такая организация позволила использовать объекты класса-потомка как отдельно, так и включая в объекты класса-предка.

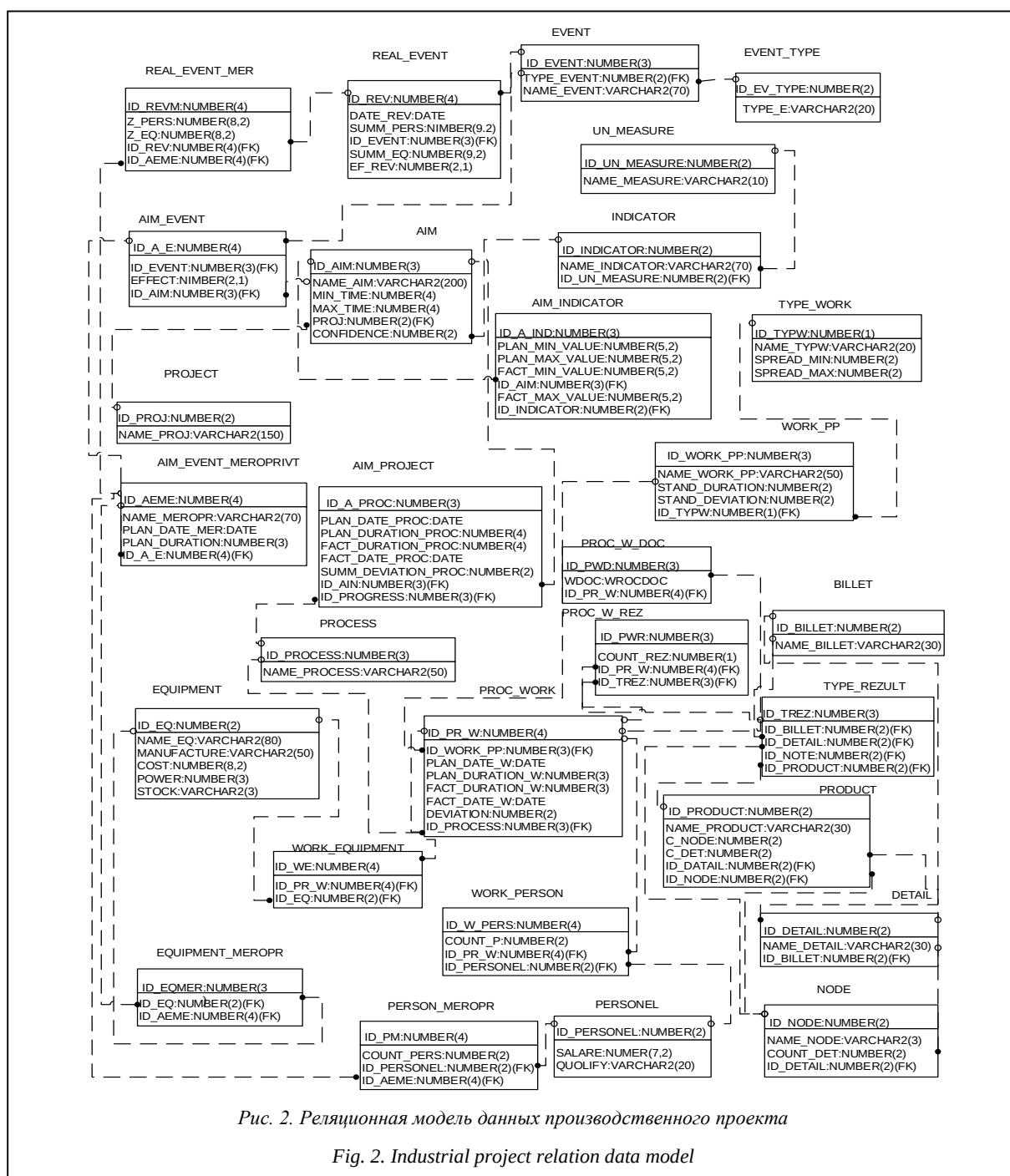
Агрегатные отношения (заканчиваются незакрашенным ромбом) реализуют логику включения одних классов в виде части целого в другой класс. Например, в классе Project (проект) должны быть установлены цели конкретного проекта, соответственно он должен включать в себя объекты класса Aim (цель). Аналогично связаны классы Aim–Process (цель–процесс), Process–Work (процесс–работа), Work–Result (работа–результат). Такое построение модели наглядно отражает логику описания проекта и гарантирует включение всех его необходимых элементов. Отношение Work–Result (работа–результат), изображенное на рисунке с закрашенным ромбом, является частным случаем агрегации – композицией. Суть данного отношения заключается в большей степени зависимости составных частей и целого, когда каждый результат выполнения тех или иных задач может быть связан только с конкретной работой.

Отношения ассоциации (сплошная линия) отражают взаимосвязь равноправных классов приложения. Цифрами над концами ассоциации на рисунке указаны диапазоны кратности ролей, отражающие количество объектов класса с данной ролью, которые могут участвовать в каждом экземпляре ассоциации.

Из рисунка 3 видно, что количество классов в данном случае меньше количества таблиц в БД, при этом пять из них – это классы-наследники, работа с которыми облегчается за счет наличия в объектном программировании механизма наследования [9]. Таким образом, применение объектного подхода к представлению информации о производственном проекте дает возможность сократить ее объем, что, в свою очередь, позволяет повысить производительность алгоритмов обработки.

Пример преобразования данных из объектной модели в реляционную

Существует несколько технологий для разработки схемы обмена данными (маппинга) между



объектами прикладного решения и таблицами реляционной БД. На основании проведенного анализа для организации преобразования объектов Java в реляционную структуру была выбрана технология JPA (Java Persistence API). Пример реализации данной технологии был выполнен с использованием библиотеки Hibernate JPA [10].

Hibernate – это библиотека, разработанная для языка программирования Java, которая обеспечивает отображение объектов в структуры реляционных данных. Использование данной библиотеки существенно сокращает время проектирования приложений, ориентированных на работу с БД,

позволяя легко организовывать связь между классами Java и таблицами БД и типами данных Java и Oracle. Кроме этого, Hibernate обладает средствами автоматического создания запросов и извлечения данных [11].

Визуально процесс взаимодействия Java и реляционных данных посредством Hibernate можно представить в виде, показанном на рисунке 4.

Полный набор файлов приложения для полноценного функционирования системы:

- description.jar – пакет классов, определяющий основные рабочие классы приложения и методы доступа к их атрибутам;

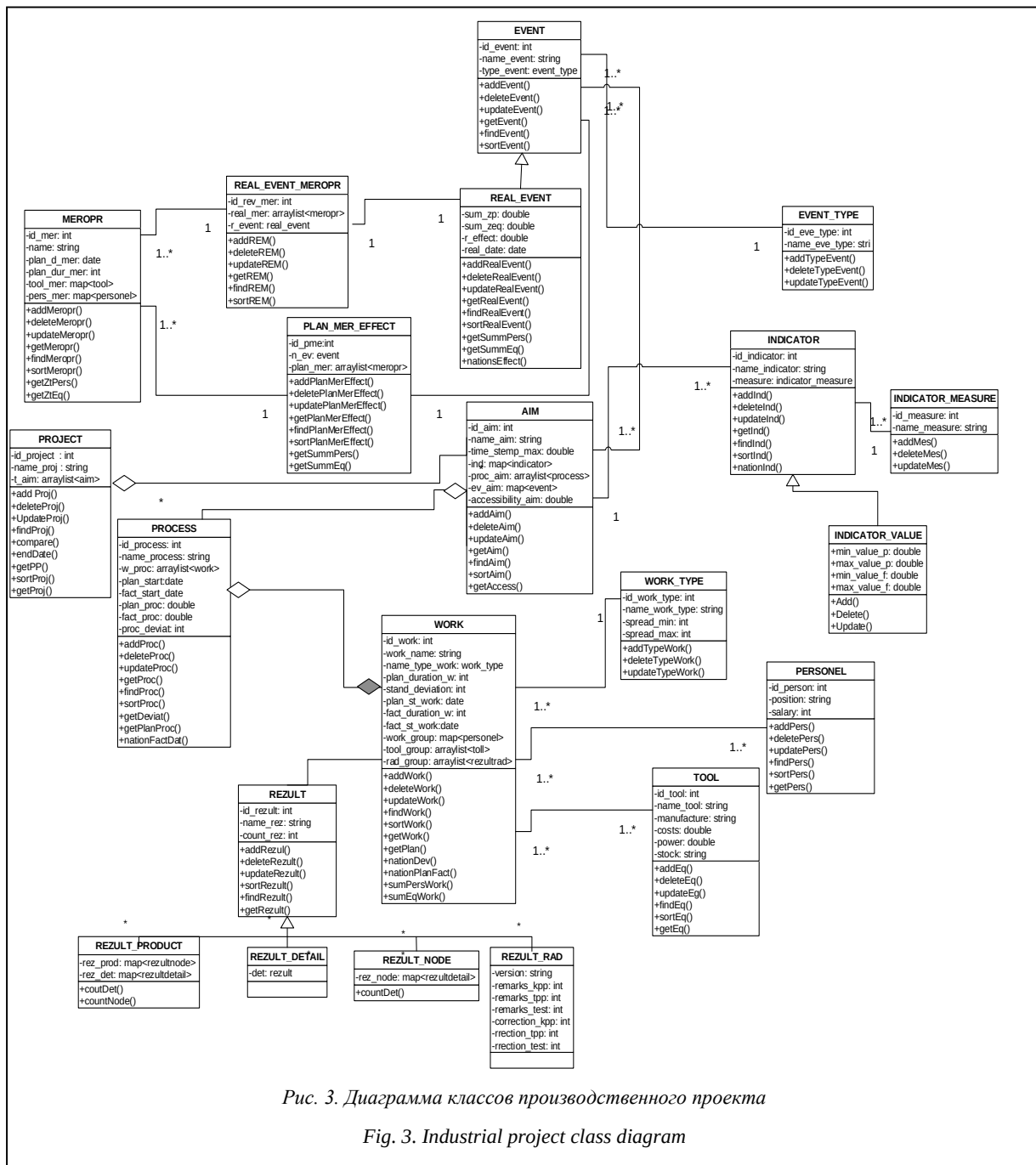


Рис. 3. Диаграмма классов производственного проекта

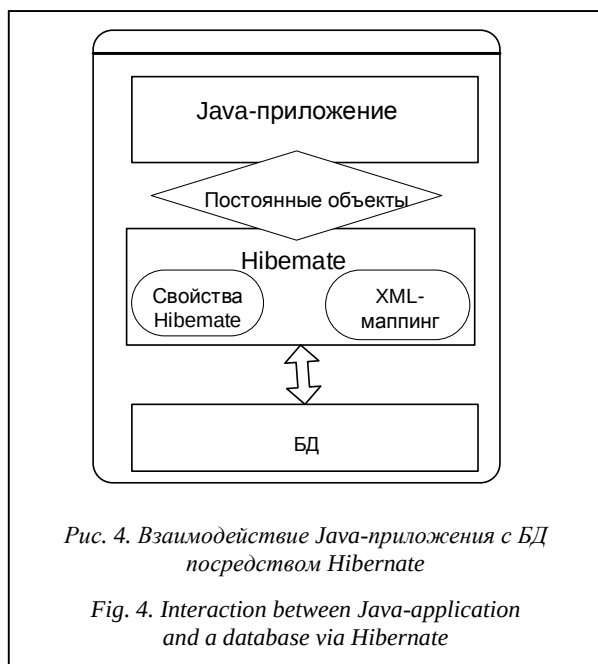
Fig. 3. Industrial project class diagram

- core.jar – набор классов, реализующих бизнес-логику приложения;
- Hibernate Library.jar – подключаемая библиотека, содержащая набор пакетов и классов для реализации технологии Object-Relational Mapping;
- condb.jar – набор файлов, отвечающих за ассоциирование java-классов приложения с таблицами БД Oracle;
- JRE System Library.jar – пакет, включающий в себя стандартный набор библиотек, необходимых для работы приложения;
- ojdbc7.jar – подключаемая библиотека, содержащая набор классов для подключения приложения к серверу с использованием драйвера JDBC.

Благодаря логической и физической группировкам отдельных элементов программы пакетная организация файлов приложения позволила обеспечить удобство навигации по коду программы. Общая структура пакетов и подключаемых библиотек приложения представлена на рисунке 5.

Пунктирными линиями на рисунке изображен процесс импортирования классов одного пакета в другой, сплошными – обращение к методам классов.

В классах пакета description методы получения и внесения данных по каждому атрибуту класса реализованы в соответствии с логикой объектно-ориентированного программирования и отвечают их



прямому назначению. Методы `set` устанавливают значения переменных классов, методы `get` получают значения и предоставляют доступ к ним из других классов, обеспечивая возможность взаимосвязанной работы модулей приложения.

Двусторонней стрелкой на рисунке отражена связь приложения с БД через драйвер JDBC. Линии, завершающиеся точками, отражают связь mapping-файлов пакета `condb.jar` с объектами приложения.

Организация связи с БД в приложении осуществляется в соответствии с технологией Object-Relational Mapping (ORM), которая предназначена для преобразования данных из реляционных СУБД в привычную для языка Java объектную форму. Реализует данную технологию механизм, описывае-

мый подключаемой библиотекой `Hibernate Library.jar` [Хемранж].

Работа с механизмом `Hibernate` включает в себя следующие этапы.

- Установка параметров соединения с БД. За соединение с БД системы отвечает конфигурационный файл `hibernate.cfg.xml` (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/5.jpg>), который содержит в себе необходимые сведения и параметры для подключения к БД и описывает связанные с базой классы. Для организации соединения используется технология `Java DataBase Connectivity (JDBC)`. Необходимый драйвер загружается из подключаемой библиотеки `ojdbc7.jar`.

- Инициализации механизма `Hibernate`.

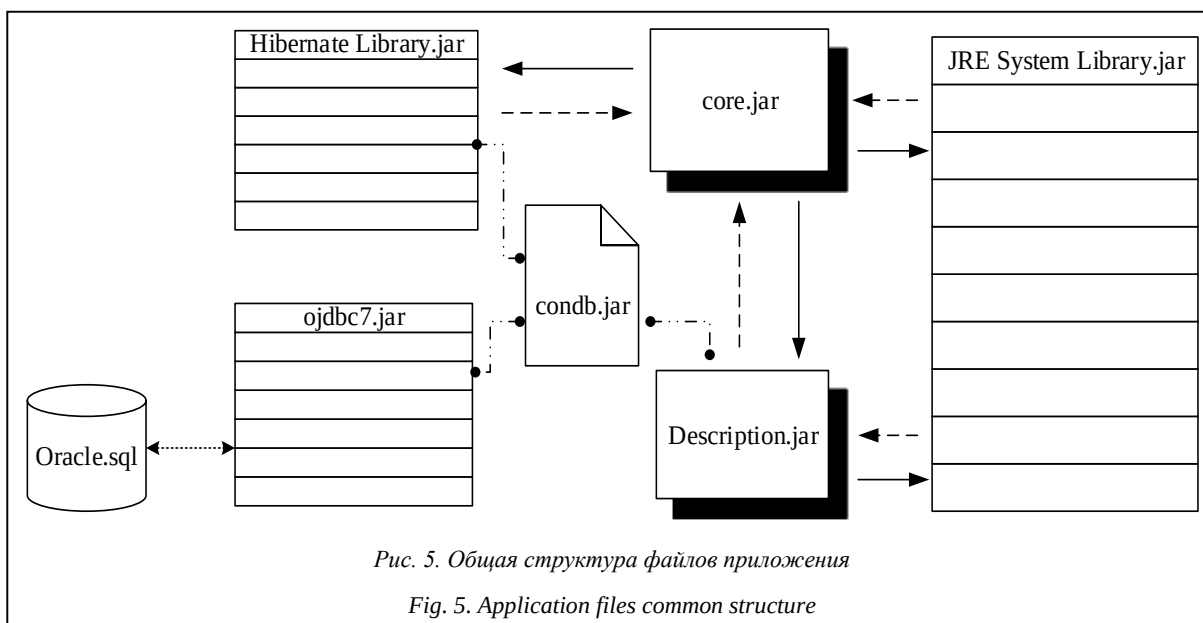
В связи с тем, что работа с БД осуществляется с помощью транзакций в пределах сессии, необходимо создать фабрику сессий (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/6.jpg>).

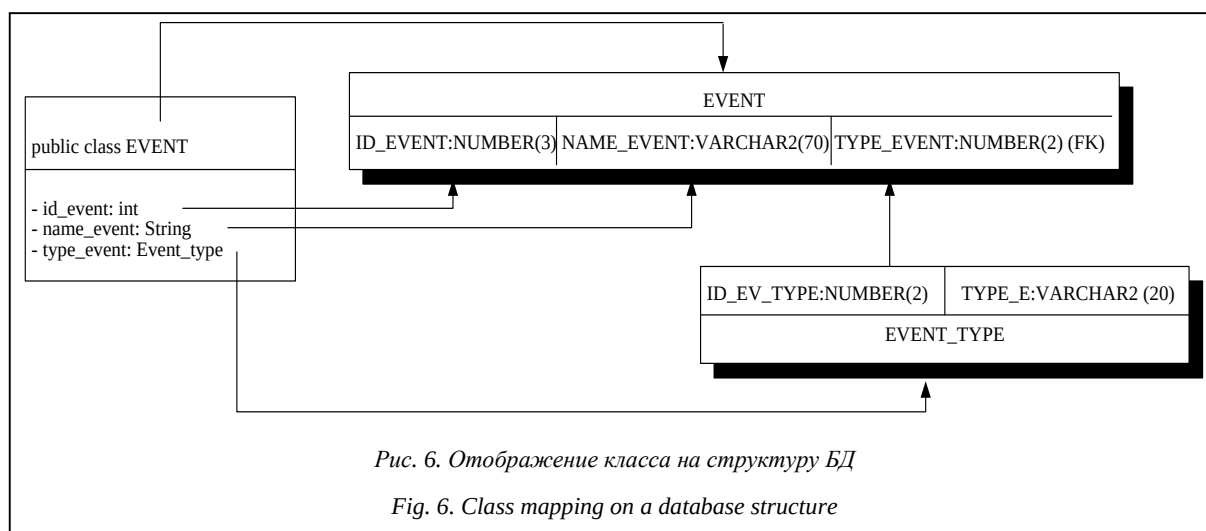
Механизм активируется с помощью служебного класса `HibernateUtil.java` (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/7.jpg>). При загрузке данного класса создается специальный объект `sessionFactory`, который считывает данные из файла `hibernate.cfg.xml` и осуществляет подключение к БД.

- Отображение классов на таблицы.
- Описание отношений между классами.

Соотношения классов приложения с таблицами БД, а также отношения между классами описываются в файлах вида `name_class.hbm.xml`. Пример отображения класса на таблицы БД представлен на рисунке 6.

В данном случае каждому атрибуту класса ставится в соответствие некоторое поле таблицы, на основании чего происходит считывание информации из БД, где каждая строка – это экземпляр некоторого класса.





Затем для организации непосредственно работы с БД создаются базовые классы DAO, которые содержат методы работы с Hibernate. Кроме этого, здесь описывается ряд аннотаций, перечень которых приведен в таблице 3.

Таблица 3

Перечень аннотаций

Table 3

Annotation list

Аннотация	Описание
@Entity	Сообщает Hibernate о взаимодействии класса с Hibernate
@Table	Явно указывает название таблицы, к которой обращаются
@Column	Указывает название атрибута таблицы, с которым осуществляется работа
@Id	Указывает первичный ключ Primary-ключа
@Generated-Value	Обеспечивает автоматическую генерацию при добавлении нового объекта в базу
@ManyToOne	Используется при работе с полем в случае, когда таблица связана с другой типом связи «один-ко-многим»
@Join-Column	Имя поля таблицы, по которому осуществляется связь «один-ко-многим» с другой таблицей
@ManyToMany	Используется при работе с полем в случае, когда таблица связана с другой типом связи «многие-ко-многим»

Использование данных классов и аннотаций в них обеспечивает быструю и простую работу с БД.

В заключение отметим следующее. Разнообразие существующих информационных систем, предназначенных для автоматизации деятельности промышленных предприятий, обеспечивает повышение эффективности бизнес-процессов за счет рационального планирования и контроля. Вместе с тем распределение процедур обработки данных между системами и самих данных между хранилищами (БД) приводит к проблемам согласованности информации и увеличению ее избыточности.

Эффективная автоматизация управления сложными производственными проектами возможна только при наличии достоверной, актуальной и непротиворечивой информации. Задача поддержания ее в таком состоянии осложняется наличием нескольких представлений: реляционного – для хранения в БД, объектного – для обработки. В статье предложены реляционная и объектная модели данных по проектам, которые могут использоваться в автоматизированных информационных системах промышленных предприятий для решения задач проектного управления. Пример преобразования данных из объектов Java в таблицы СУБД Oracle, выполненный с применением библиотеки Hibernate, служит иллюстрацией преобразования моделей данных различных типов. Он может быть взят за основу при разработке модулей обмена данными между реляционными БД и прикладными программами, созданными с помощью объектных языков программирования.

Литература

1. Загидуллин Р.Р. Управление машиностроительным производством с помощью систем MES, APS, ERP. Старый Оскол: ТНТ, 2011. 372 с.
2. Фролов Е.Б., Загидуллин Р.Р. MES-системы, как они есть, или эволюция систем планирования производства // Металлообрабатывающее оборудование. 2008. № 10 (55). С. 31–37.
3. Стоянова О.В. Оценка соответствия существующих информационных систем управления проектами особенностям проектного управления в наноиндустрии // Программные продукты и системы. 2013. № 3. С. 193–199.
4. Палюх Б.В., Бурдо Г.Б. Метод интеллектуальной оценки решений при проектировании технологии в многоменклатурных производствах // Вестн. Тамбовского гос. техн. ун-та. 2011. Т. 17. № 2. С. 342–350.
5. Бурдо Г.Б., Стоянова О.В. Автоматизированная система управления процессами создания наукоемких машиностроительных изделий // Программные продукты и системы. 2014. № 2. С. 164–170.
6. Гришина Т.Г. Моделирование и оптимизация циклов выработки решений при управлении автоматизированным производством // Инженерный вестник Дона. 2012. № 3 (21). С. 390–394.
7. Андреев В.В., Тесленко Е.В., Хрилов В.П. Динамиче-

ская модель управления конструкторско-технологическим взаимодействием в САПР // Науч.-технич. вестн. Поволжья. 2013. № 3. С. 64–67.

8. Туровец О.Г., Родионов В.Б., Бухалков М.И. Организация производства и управление предприятием. М.: Инфра-М, 2013. 3-е изд. 506 с.

9. Хабибуллин И.Ш. Java 7: Наиболее полное руковод-

ство. СПб: БХВ-Петербург, 2012. 768 с.

10. Хемрадхани А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse; [пер. с англ. В. Коваленко]. М.: Вильямс, 2008. 352 с.

11. Браусоу К. Hibernate упрощает преобразование наследования. IBM developerWorks. URL: <http://www.ibm.com/developerworks/ru/library/j-hibernate/> (дата обращения: 28.09.2015).

DOI: 10.15827/0236-235X.112.210-218

Received 30.09.15

COMPLEX INDUSTRIAL PROJECTS DATA REPRESENTATION MODELS IN AUTOMATED INFORMATION SYSTEMS OF INDUSTRIAL INSTITUTIONS

(The work has been done with financial support from RFBR within the research projects no. 15-07-02935 A)

Dli M.I., Dr.Sc. (Engineering), Professor, midli@mail.ru;

Stoyanova O.V., Ph.D. (Economics), Associate Professor, ovstoyanova@list.ru

(Smolensk Branch of the Moscow Power Engineering Institute,

Energeticheskii proezd 1, Smolensk, 214013, Russian Federation);

Belozersky A.Yu., Dr.Sc. (Economics), Professor

(D. Mendeleev University of Chemical Technology of Russian Federation,

Miuskaya Sq. 9, Moscow, 125047, Russian Federation)

Abstract. A great number of various information systems is used during the process of complex industrial projects management. Each of them solves independent projects data processing tasks. Thus, as for new production setup projects, product information is distributed between computer-aided design systems (CAD) and automated process control systems (APCS), information about necessary resources is distributed between APCS, management information system (MIS), automated project management system (APMS), etc. Project data exchange between stated systems is complicated by using different hardware and software, data formats, data representation models. The article considers object and relation data representation models of complex industrial projects. The first type of model is used as base of data processing using object-oriented programming languages. The relation model is necessary for data storage in industrial institutions databases. This situation is caused by information technologies evolution, i.e. data processing application development using object-oriented programming and relation DBMS dominance. Different units of information systems use different data models. This requires decision-making on transforming data representations. There is no one-stop solution for the stated task for now. The paper considers the example of such transformation for an object model in Java and for a relation model in DBMS Oracle.

Keywords: industrial project, project data model, project management, computer-based information service.

References

1. Zagidullin R.R. *Upravlenie mashinostroitelnyim proizvodstvom s pomoshchyu sistem MES, APS, ERP* [Machine-building industry management using MES, APS, ERP]. Stary Oskol, TNT Publ., 2011, 372 p.
2. Frolov E.B., Zagidullin R.R. MES as they are or production scheduling systems evolution. *Metalloobrabatyvayushchee oborudovanie* [Metal-working Equipment]. 2008, no. 10 (55), pp. 31–37 (in Russ.).
3. Stoyanova O.V. Project management information systems and their using in the field of nanoindustry. *Programmnye produkty i sistemy* [Software & Systems]. 2013, no. 3, pp. 193–199 (in Russ.).
4. Palyuch B.V., Burdo G.B. Intellectual decisions estimation method in technology design of multiproduct productions. *Vestn. Tambovskogo gos. tekhnich. un-ta* [Transactions of the TSTU] 2011, vol. 17, no. 2, pp. 342–350 (in Russ.).
5. Burdo G.B., Stoyanova O.V. Automated control system for science-based mechanical engineering product development. *Programmnye produkty i sistemy* [Software & Systems]. 2014, no. 2, pp. 164–170 (in Russ.).
6. Grishina T.G. Modeling and optimization of decisions making cycle in automated production. *Inzhenerny vestnik Dona* [Engineering Journ. of Don]. 2012, no. 3 (21), pp. 390–394 (in Russ.).
7. Andreev V.V., Teslenko E.V., Khranilov V.P. Dynamic management model of CAD design-technological interaction. *Nauch.-tekhnich. vestn. Povolzhya* [Scientific and Technical Volga region Bulletin]. 2013, no. 3, pp. 64–67 (in Russ.).
8. Turvets O.G., Rodionov V.B., Bukhalkov M.I. *Organizatsiya proizvodstva i upravlenie predpriyatiem* [Production Organization and Enterprise Management]. 3rd ed. Moscow, Infra-M Publ., 2013, 506 p.
9. Khabibullin I.Sh. *Java 7: Naibolee polnoe rukovodstvo* [Java 7: the Most Complete Manual]. St.-Petersburg, BHV-Peterburg Publ., 2012, 768 p.
10. Khemrajani A. *Agile Java Development with Spring, Hibernate and Eclipse*. Sams Publ., 2006, 360 p. (Russ. ed.: V. Kovalenko, Moscow, Vilyams Publ., 2008, 352 p.).
11. Coulon X., Brousseau Ch. *Hibernate simplifies inheritance mapping*. Available at: <http://www.ibm.com/developerworks/java/library/j-hibernate/> (accessed September 28, 2015).

УДК 004.94:519.688

Дата подачи статьи: 01.06.15

DOI: 10.15827/0236-235X.112.219-230

ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ПРОГНОЗИРОВАНИЯ ПОТРЕБИТЕЛЬСКОГО СПРОСА И ОПТИМИЗАЦИИ ТОВАРНЫХ ЗАПАСОВ

А.А. Грицай, аспирант, allgrit@forecastnow.ru

(Тверской государственный университет, ул. Желябова, 33, г. Тверь, 170100, Россия)

В статье представлена автоматизированная система управления и оптимизации товарных запасов торговых предприятий. Система ориентирована на малые и средние торговые предприятия. Она позволяет высвобождать денежные средства в размере 20–30 % от объема запасов и увеличивать уровень удовлетворенного спроса на 5–7 %. Это достигается за счет точного прогнозирования товарного запаса исходя из оптимального уровня сервиса для каждой товарной позиции. В статье описаны основные возможности системы: предобработка данных, прогнозирование спроса, расчет оптимального уровня сервиса, расчет оптимального товарного запаса, расчет заказа поставщику. Предобработка данных производится для очистки данных от резких выбросов, воздействия маркетинговых акций, влияния праздников и отсутствия товара на складе. Прогнозирование спроса осуществляется на основе временного ряда, прошедшего предобработку. Для каждого товара вычисляются характеристики его временного ряда истории продаж. В зависимости от характеристик выбирается алгоритм прогнозирования. Страховой запас определяется с заданным уровнем сервиса. Оптимальный уровень сервиса рассчитывается исходя из стратегии максимизации рентабельности вложений торгового предприятия. На основании рассчитанного уровня сервиса и выбранного метода прогнозирования производится расчет оптимального товарного запаса. С учетом имеющегося товара на складе предприятия, товара в пути, а также резервов под конкретных покупателей вычисляется оптимальный размер заказа по данной товарной позиции. После этого применяются позиционные ограничения на заказ (например кратность упаковки), а затем ограничения на заказ поставщику в целом (например весогабаритные). В работе приведен пример использования системы и сделан вывод об экономическом эффекте от ее внедрения на реальном торговом предприятии.

Ключевые слова: управление товарными запасами, оптимизация товарных запасов, система управления запасами, нейронные сети, прогнозирование спроса, оптимальный уровень сервиса, заказ поставщику.

В связи с состоянием экономики России потребность в решении задач оптимизации бизнес-процессов растет с каждым годом. Это происходит из-за снижения нормы прибыли во многих отраслях.

В настоящий момент на оптово-розничных торговых предприятиях существует проблема неэффективного использования оборотных средств. Основным видом использования оборотных средств для предприятий торговли являются товарные запасы. Для нормального функционирования торгового предприятия необходимо поддерживать запасы по складским позициям для обеспечения покупательского спроса. Отсюда возникает задача определения оптимального запаса по товарной позиции на определенный период и момента заказа.

На малых и средних торговых предприятиях решать ее эффективно не всегда возможно. Это связано с недостаточной адекватностью используемых инструментов и большой сложностью более эффективных математических моделей и алгоритмов. В результате по одним товарным позициям возникает большой излишек, в то время как по другим значительный дефицит. В итоге предприятие страдает от потерь из-за отсутствия товара, необходимого покупателям, упущенного спроса и от потерь денежных средств, замороженных в избыточных объемах запасов и списаний по срокам годности.

Существующие разработки для малого и среднего бизнеса либо имеют скудные возможности, низкое качество прогноза и плохую поддержку,

либо являются сложными, громоздкими, дорогостоящими системами с большими сроками внедрения и высоким порогом для использования.

В данной работе предлагается программный продукт, не имеющий этих недостатков и позволяющий эффективно управлять товарными запасами.

Возможности системы

Программный продукт Forecast NOW! [1, 2] предназначен для прогнозирования потребительского спроса и оптимизации товарных запасов торговых компаний со своим складом за счет автоматизации процесса расчета заказа поставщику и увеличения точности рассчитанной потребности в товаре. Продукт ориентирован, прежде всего, на малые и средние торговые предприятия с числом товарных наименований от 1 тысячи уникальных позиций и годовым оборотом от 10 млн до 1 млрд рублей.

Система позволяет оптимизировать складские товарные запасы, снизить издержки на их хранение, повысить оборот денежных средств предприятия за счет сокращения излишних запасов, повысить объем продаж за счет увеличения уровня удовлетворенного спроса (сокращения дефицита), существенно сократить время на формирование заказа поставщику, автоматизировать работу отдела закупок.

Для решения этих задач строятся прогнозы спроса на каждую товарную позицию. При постро-

ении прогноза спроса учитываются характеристики временного ряда спроса, частота продаж, их вариативность. Для каждого товара выбирается оптимальный метод прогнозирования согласно дереву решений в зависимости от рассчитанных параметров. Среди используемых методов прогнозирования – композиции нейронных сетей [3] и генетических алгоритмов [4], методы математического моделирования. Опираясь на построенные ранее прогнозы, осуществляется расчет оптимального товарного запаса, потребности в товаре с учетом требуемого уровня сервиса, частоты поставок, кратности партии, минимального остатка на полке и прочих ограничений. На основе рассчитанных потребностей формируется отчет о необходимых закупках, который может быть предоставлен как по товарным группам, так и в разрезе поставщиков или производителей.

Программная система может формировать заказы поставщику, распределительному центру или

рассчитывать оптимальные внутренние перемещения [5].

Разработаны специализированные версии системы для различных отраслей торговли [6].

Общая архитектура и интерфейс системы

Проект состоит из двух основных частей: библиотеки Core.lib («Ядро»), содержащей в себе функционал для взаимодействия с БД с проведением анализа данных и расчетный блок, а также графического интерфейса пользователя. В процессе создания интерфейса пользователя использован стандартный подход разделения объектов на модели и представления, обогащенный предоставляемой библиотеками Qt системой слотов и сигналов. Общая схема действий пользователя (рис. 1) отражает главные компоненты интерфейса пользователя, их можно разделить на четыре группы: расчет и сохранение заказа (рис. 2), проведение анали-

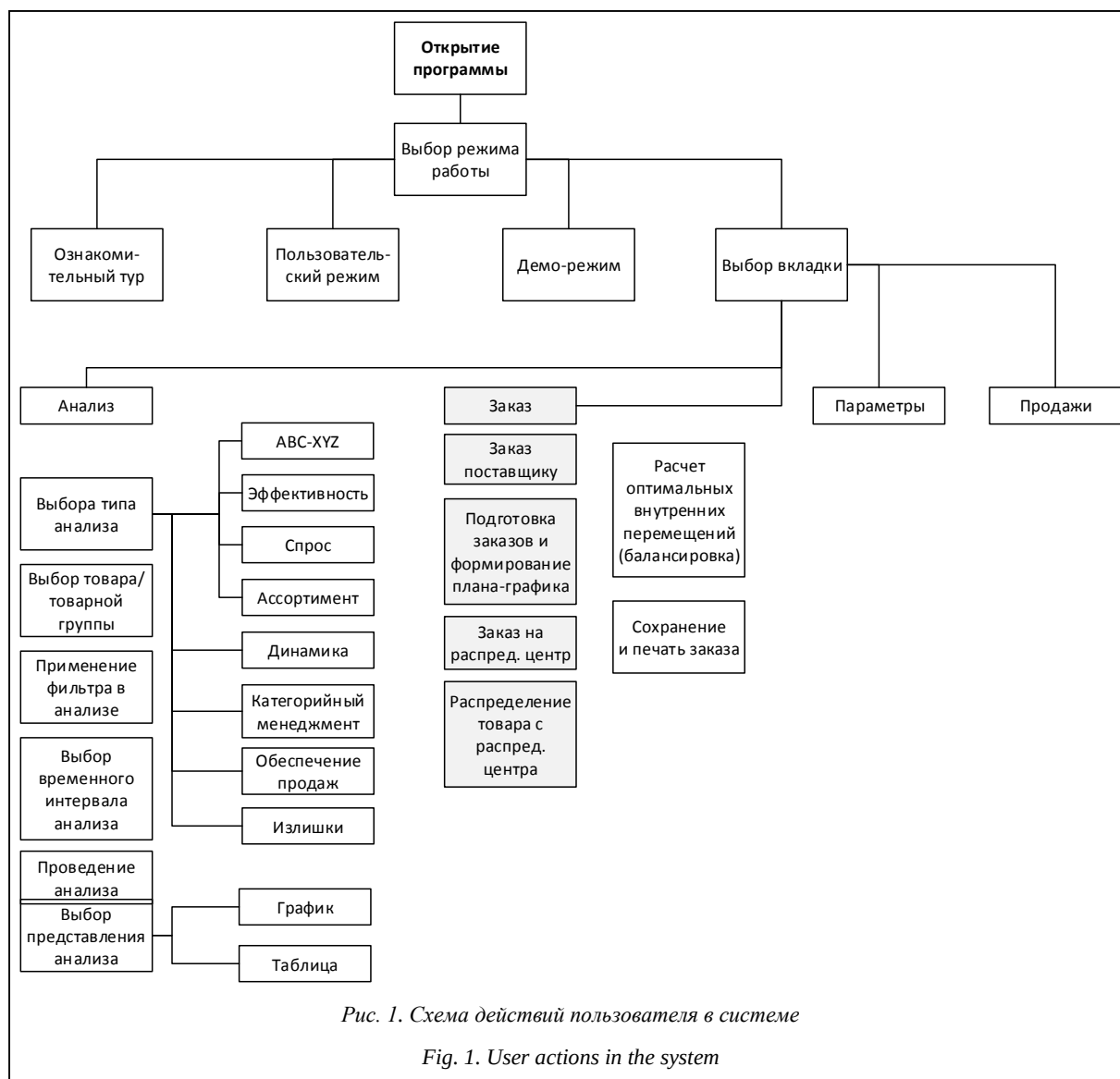


Рис. 1. Схема действий пользователя в системе

Fig. 1. User actions in the system

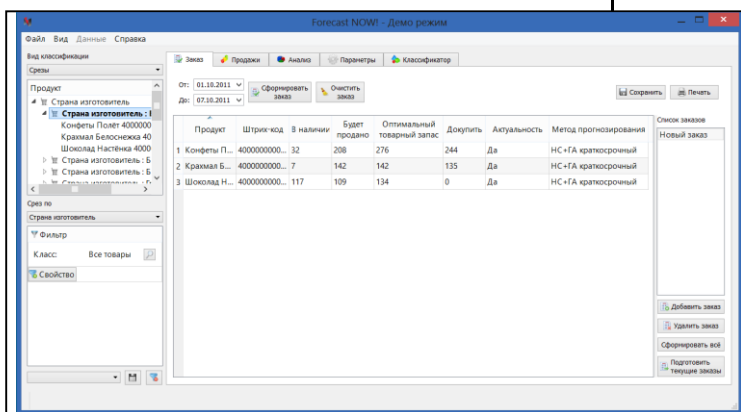


Рис. 2. Расчет и сохранение заказа

Fig. 2. Order calculation and saving

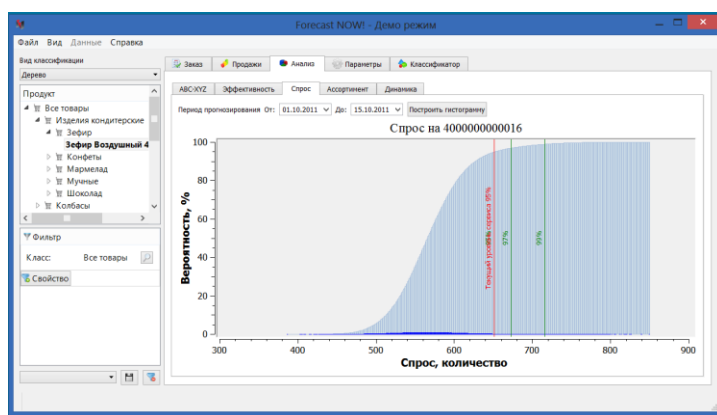


Рис. 3. Анализ распределения спроса на товарную позицию

Fig. 3. Product item demand distribution analysis

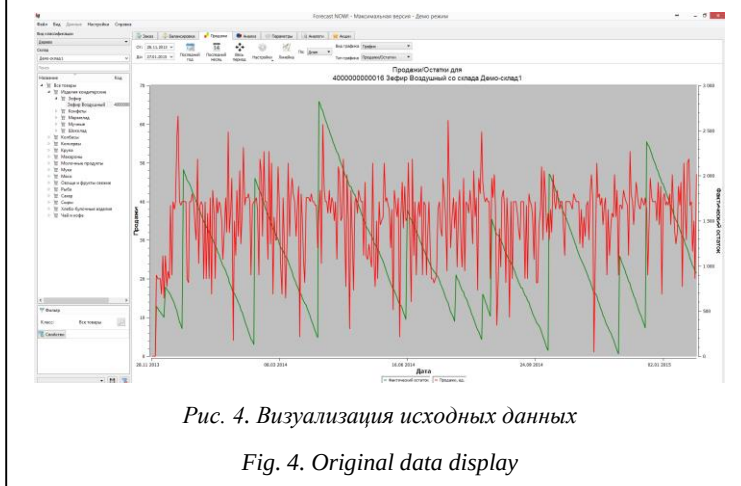


Рис. 4. Визуализация исходных данных

Fig. 4. Original data display

зов (рис. 3), настройка параметров, визуализация исходных данных (рис. 4). Визуально эти группы разделены между собой отдельными вкладками.

База данных

Для хранения данных используется встроенная БД SQLite3. Выбор данной СУБД обусловлен тем,

что она обеспечивает максимальную производительность на ограниченных объемах данных в однопользовательском режиме, который подразумевает использование одной БД одним пользователем. Это не является ограничением для аналитической системы: первичные данные находятся только в учетной системе пользователя, а синхронизация осуществляется посредством обмена данными с этой системой.

Общая схема таблиц БД (рис. 5) содержит несколько основных таблиц. Таблица sales предназначена для хранения истории продаж и остатков, получена из учетной системы пользователя. Таблица productsTable хранит информацию о товарах, складах (торговых точках) и их уникальных идентификаторах. Таблица с аналогами analogs предназначена для работы механизма учета аналогичных товаров и их продаж при формировании прогноза и расчета потребности в товаре. Учет маркетинговых акций и привязка к товарам осуществляются с использованием таблиц campaigns и itemCampaigns. Классификация товаров, а также их свойства реализованы на основе троек <Субъект, Предикат, Объект> и хранятся в таблице Triples. Рабочие и выходные дни определяются таблицами schedule и scheduleWeekType. Информация о транспортных средствах с их весогабаритными характеристиками находится в таблице carriers. Для разграничения прав пользователей реализован соответствующий механизм, использующий таблицу users.

Ядро системы

Ядро системы Forecast NOW! включает несколько блоков (рис. 6). Данные импортируются из корпоративной информационной системы (КИС) пользователя, затем проходят проверку на корректность. После этого согласно заданному расписанию поставок форми-

руется план-график заказов, которые необходимо рассчитать. Подготовленный список заказов поступает на вход модуля «Формирование заказов», который рассчитывает необходимый товарный запас по каждой товарной позиции и определяет потребность в соответствии с текущими остатками товара, резервами, товаром в пути. Модуль «Учет ограничений» позволяет учесть такие ограничения на за-

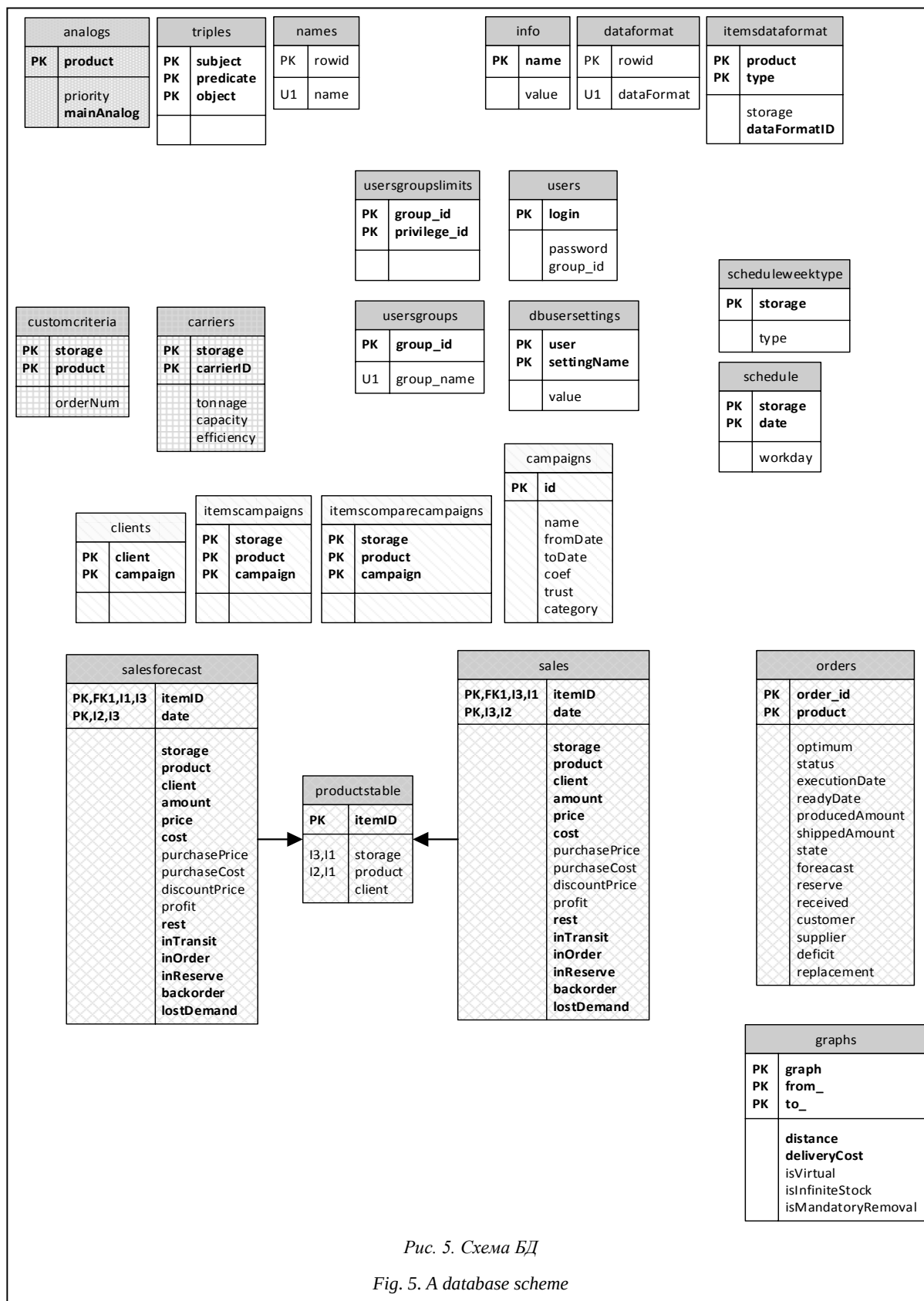


Рис. 5. Схема БД

Fig. 5. A database scheme

каз, как минимальная партия, кратность упаковки и пр. В результате формируются рассчитанные заказы, которые могут быть экспортированы в удоб-

ном формате (xls, csv, html). После этого заказ может быть отправлен поставщику или загружен в КИС пользователя.

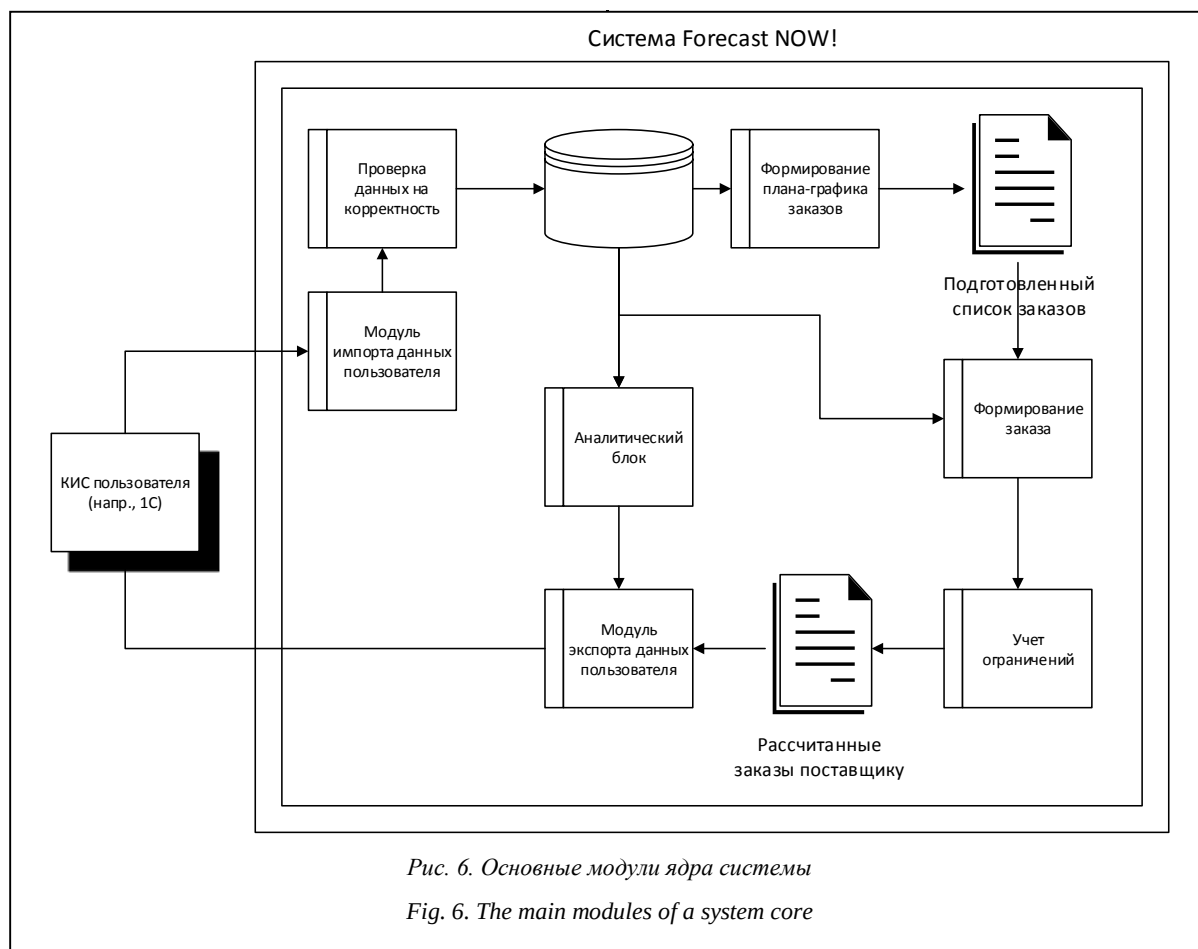


Рис. 6. Основные модули ядра системы

Fig. 6. The main modules of a system core

Модуль импорта данных

Разработанная система использует ряд внешних данных, в число которых в обязательном порядке входят данные об истории продаж, остатков и цен закупки и реализации. В качестве источников данных используются csv-файлы гибкой структуры. Пользователь может выбрать порядок столбцов, разделители, формат даты, целой и дробной частей.

Для сохранения истории продаж в csv-файл пользователи программного продукта «1С: Предприятие» могут применять специально созданную внешнюю обработку.

Кроме обязательных данных, система также может импортировать данные о классификации товаров, их параметрах, аналогах и маркетинговых акциях.

История продаж. Включает следующие данные о товаре: продажи, фактический остаток, товар в пути, товар в резерве, размещенные заказы поставщику, цены закупки и реализации. Данные выгружаются ежедневно, если было какое-то их изменение.

Тройке (уникальный идентификатор товара, склад, дата) соответствует одна строка.

Классификация товаров. Содержит информацию о свойствах и признаках товара, их группах и

подгруппах и о принадлежности товара той или иной группе.

Необходимо загружать программу при изменениях классификации (добавляется новый товар или свойство, меняется значение свойства и т.п.).

Параметры. В программе используются различные параметры, такие как

- срок годности товара в днях: применяется при оценке рисков потерь от списания испорченной продукции;
- маркер «Нового товара»: сколько дней с момента первой продажи товар считается новым (не рассчитывается заказ);
- минимальный запас, необходимый для поддержания на складе вне зависимости от спроса на товар.

Аналоги. Содержит информацию об аналогичных товарах. Задаются главный и младшие аналоги. В заказ или анализ будет попадать главный аналог. При этом суммируется история продаж и остатков по товарам-аналогам. При попытке добавления младшего аналога в заказ или анализ будет добавлен главный аналог.

Акции. Содержит информацию об акциях, такую как период акции, ожидаемый коэффициент роста продаж по акции, список товаров, участвующих в акции.

Проверка данных на корректность

Для корректной работы системы необходимо обеспечить поступление корректных данных на вход. Для этого производится анализ данных, поступающих от пользователя в формате csv.

При импорте данных пользователем история продаж и остатков проверяется на наличие следующих проблем:

- 1) не задано какое-либо из полей;
- 2) неверный формат полей;
- 3) неверное количество полей;
- 4) цена закупки выше или равна цене реализации;
- 5) цена закупки равна нулю, при этом продажи больше нуля;
- 6) противоречивые данные (несколько разных строк) для тройки «склад, товар, дата»;
- 7) несколько разных названий товара для одного кода;
- 8) одинаковые строки в выгрузке;
- 9) наличие разрывов в истории продаж после импорта выбранных данных;
- 10) более 50 % остатков равны нулю.

Проблемы делятся на две категории – критичные и предупреждения. При обнаружении критичных проблем дальнейший импорт невозможен. Пример критичной проблемы – 1, 2, 3. При обнаружении не критичных проблем выводятся предупреждения с указанием строки и места в строке, где данная проблема обнаружена, а также возможные способы устранения данной проблемы. Например, для проблемы 6 это рекомендация проверить группировку данных по дням.

Производятся два вида проверок исходных данных – синтаксические и семантические. Синтаксические проверки осуществляются без рассмотрения предметной области и ее ограничений (пример – 1, 2, 3, 8). Семантические правила опираются на экспертные знания из предметной области.

Приведем пример работы одного правила.

Правило «Наличие разрывов в истории продаж после импорта выбранных данных» должно обеспечить отсутствие пропусков в истории продаж и обрабатывается следующим образом.

Если количество рабочих дней между максимальной датой в базе и минимальной датой в загружаемых данных больше нуля, то выводится предупреждение следующего вида. *История продаж в базе загружена по МАКС_ДАТА(База). Пользователь загружает данные о продажах с МИН_ДАТА(История) по МАКС_ДАТА(История). За период с МАКС_ДАТА(База)+1 по МИН_ДАТА(История) данные о продажах отсутствуют. Если это выходные или праздничные дни, то в этом нет ничего страшного. В другом случае вам следует загрузить данные за пропущенный период, иначе составленные прогнозы будут недостоверны.*

Здесь МАКС_ДАТА(База) – последняя дата, на которую информация по истории продаж и остатков сохранена в базе; МИН_ДАТА(История) – первая дата в загружаемой истории продаж и остатков; МАКС_ДАТА(История) – последняя дата в загружаемой истории продаж и остатков.

Формирование плана-графика заказов

План-график заказов формируется на основе расписания заказов, которое может быть задано в виде календаря с отмеченными днями возможного заказа, а также с помощью начальной даты отсчета и периодичности заказов. Первый вариант позволяет реализовывать сложные схемы расписания заказов, например, каждые вторник и среду по нечетным неделям месяца. Второй вариант более простой и позволяет задавать расписание следующего вида: раз в неделю по вторникам. Расписание заказов может быть уникальным для каждой пары товар–склад. Это позволяет очень гибко определять правила заказов товаров в зависимости от склада, поставщика или любого другого свойства товара.

Помимо даты размещения заказа, для расчета потребности необходимо знать горизонт планирования (на какой период этим заказом планируется гарантированно удовлетворить спрос с заданным уровнем сервиса), а также срок поставки (время с момента размещения заказа до момента его поступления на склад и возможности его реализации покупателям). Эти параметры также могут быть заданы для каждой пары товар–склад уникальными или в зависимости от поставщика или любого другого свойства.

В результате система определяет, какие товары могут быть сегодня заказаны, и группирует их согласно следующему кортежу: склад–заказчик, поставщик, дата размещения заказа, дата поступления заказа, плановая дата окончания товарного запаса.

Таким образом получается список заказов, состоящий из списка товаров, по которым необходимо рассчитать потребность на даты (дата поступления заказа, плановая дата окончания товарного запаса), а также расход до поступления товара на даты (текущая дата, дата поступления заказа).

После формирования плана-графика производится расчет потребности в товаре для каждого товара в каждом заказе. В результате получаются рассчитанные заказы поставщику с учетом ограничений поставки на конкретную позицию. Ограничения на весь заказ позволяет учесть модуль «Учет ограничений». Для оценки эффективности управления товарными запасами предназначен аналитический блок. Он позволяет также принимать решения относительно уровня сервиса, который рентабельно обеспечивать компании по той или иной товарной позиции.

Модуль формирования заказа

Основной задачей системы является расчет заказов для поставщиков, осуществляемый модулем формирования заказов. Как было отмечено ранее, на вход этого модуля подается календарный план-график заказов, сформированный на основе расписания поставок. Задачей модуля формирования заказов (рис. 7) является определение объемов заказа по каждой из позиций плана-графика.

Предобработка истории продаж. Для получения точного расчета оптимального товарного запаса необходима качественная предобработка исходных данных. Исходные данные могут иметь следующие особенности:

- неслучайное отсутствие спроса в моменты дефицита;
- завышенный спрос в моменты маркетинговых акций;
- заниженный спрос в постмаркетинговый период;
- аномально большие нехарактерные продажи.

В большинстве случаев фактор качественной предобработки является решающим и позволяет увеличить точность прогнозирования на 10–15 % при наличии незначительной предобработки и на 30–40 % при полном изначальном ее отсутствии.

Важным этапом предобработки данных является фильтрация, так как она позволяет сгладить временной ряд, уменьшить число выбросов, снизить содержание шума в выборке. Фильтрация мо-

жет действовать локально, на конкретные выбросы, а также глобально – на весь временной ряд.

Для решения первой задачи используются анализ частот объемов продаж и отсечение по заранее заданному порогу. Также используется ряд критериев локальной фильтрации – фильтрация выбросов использованием критериев Диксона [7, 8], Шовене [9], Титьена–Мура [10]. Статистические критерии позволяют проверить гипотезу о наличии значительного отклонения в данных, которое носит случайный характер и с большой вероятностью является шумом.

Для анализа с использованием критерия Диксона исходный ряд преобразуется в возрастающую вариативную последовательность с повторениями $x_1, \dots, x_n$, а сам критерий определяется отношением

$$\frac{x_n - x_{n-1}}{x_n - x_1} > X.$$

Гипотеза проверяется с использованием известных значений для критерия Диксона.

Критерий Шовене [9, 11] базируется на подсчете ожидаемого числа значений $n_{\text{ожид}}$, которые имеют такую же погрешность, как и тот, в отношении которого проверяется гипотеза. Для проверки гипотезы вычисляются выборочное среднее и среднеквадратичное отклонения. Согласно таблице нормального распределения определяется вероятность появления подозрительного элемента P_z . Тогда для выборки размером n ожидается появление $n_{\text{ожид}}$ с подобной погрешностью согласно $n_{\text{ожид}} = n \times P_z$.

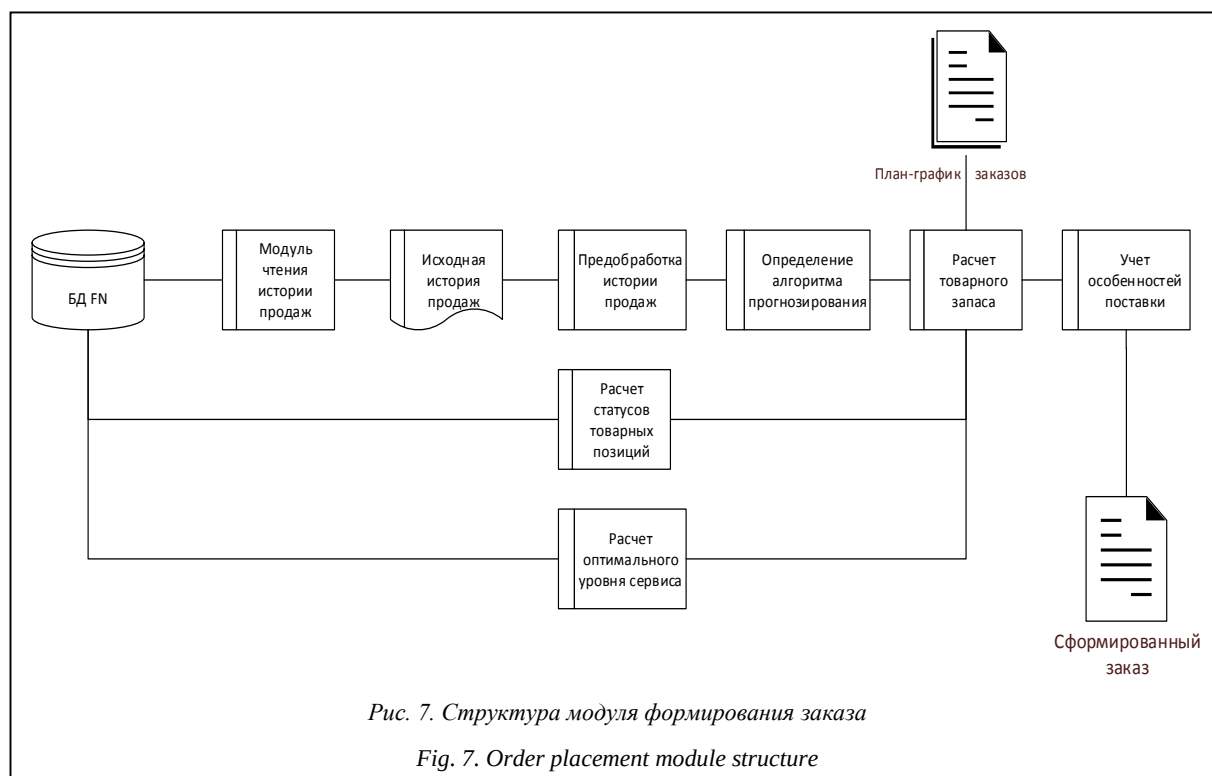


Рис. 7. Структура модуля формирования заказа

Fig. 7. Order placement module structure

Для проверки гипотезы согласно критерию Тьюэна–Мура [10] производится вычисление статистики $L_{(k)}$ для k минимальных значений выборки:

$$L_{(k)} = \frac{\sum_{i=1}^{n-k} (x_{(i)} - \bar{x}_k)^2}{\sum_{i=1}^n (x_{(i)} - \bar{x})^2}.$$

Далее осуществляется сравнение с известными критическими значениями, на основании чего гипотеза или опровергается, или принимается.

Глобальная фильтрация основана на преобразовании Фурье, которое позволяет перейти от значений временного ряда к его составляющим – гармоническим колебаниям различной частоты. Шумовая компонента является высокочастотной. Исключая колебания таких частот из сигнала и производя обратное преобразование, можно понизить уровень шума в исходном сигнале.

Расчет товарного запаса. Оптимальный товарный запас рассчитывается на основании прогноза спроса и страхового запаса на случай колебания спроса или неточности прогноза [12]. Альтернативным подходом, используемым для расчета оптимального запаса, являются методы математического моделирования и эмпирического восстановления распределения вероятности спроса [13]. Решение о выборе алгоритма прогнозирования принимается автоматически на основе дерева решений исходя из характеристик временного ряда спроса (среднего интервала между фактами продаж, вариативности этих интервалов, вариативности объемов продаж) [14].

Расчет заказа поставщику. Для расчета заказа поставщику нужно знать оптимальный товарный запас, который необходим для обеспечения спроса с заданным уровнем сервиса в течение периода между поставками, а также прогнозируемый остаток на момент поступления на склад рассчитываемого заказа.

Прогнозируемый остаток определяется по следующей формуле:

$$\text{forecastStock} = \max([\max(\text{actual} - \text{reserve}; 0) + \text{transit} + \text{ordered} - \text{consumption}]; 0),$$

где *actual* – фактический остаток на настоящее время; *transit* – количество товара, отгруженное поставщиком, но не поступившее на склад; *ordered* – количество товара, заказанное у поставщика, но еще не отгруженное; *consumption* – расчетный расход до поступления; *reserve* – зарезервированное количество товара. Первые четыре параметра берутся из учетной системы, а расход до поступления рассчитывается системой.

Расход до поступления – количество товара, необходимое для удовлетворения спроса с заданной вероятностью, на период с последней даты в базе до даты поступления рассчитываемого заказа.

На расход до поступления, в свою очередь, влияют выставленные акции, сезонность, тренд и прочие факторы.

В случае фиксированного расписания поставок количество товара, которое необходимо заказать, рассчитывается следующим образом:

$\text{orderAmount} = \max(0; \text{optimalStock} - \text{forecastStock})$, где *orderAmount* – количество товара для заказа; *optimalStock* – оптимальный запас; *forecastStock* – прогнозируемый остаток товара.

Теперь рассмотрим, как рассчитывается количество товара, которое необходимо заказать в случае наличия точки заказа. Под точкой заказа подразумевается размер товарного запаса в натуральных единицах или днях покрываемого спроса. Если прогнозируемый остаток опускается ниже точки заказа, производится заказ по этой позиции, в противном случае – нет:

Если $\text{forecastStock} < \text{orderPoint}$,
 $\text{orderAmount} = \max(0; \text{optimalStock} - \text{forecastStock})$,
 иначе $\text{orderAmount} = 0$,

где *orderPoint* – точка заказа, нат. ед.; *orderAmount* – объем заказа; *optimalStock* – оптимальный запас; *forecastStock* – прогнозируемый остаток.

Расчет оптимального уровня сервиса. Под оптимальным уровнем сервиса подразумевается такой уровень сервиса, для обеспечения которого суммарные потери денежных средств минимальны.

Таким образом, оптимальным уровнем сервиса считается такой уровень сервиса, объем запасов *vl* при котором является решением экстремальной задачи:

$$\text{Costs}(vl) = \text{Storage}(vl) + \text{MoneyC}(vl) + \text{DeficitC}(vl) + \text{ExpiredC}(vl)$$

$$\text{Costs}(vl) \xrightarrow{vl} \min$$

при ограничениях: $\text{StorageC}(vl) \geq 0$, $\text{MoneyC}(vl) \geq 0$, $\text{DeficitC}(vl) \geq 0$, $\text{ExpiredC}(vl) \geq 0$, $vl \geq 0$.

При расчете функционала потерь учитываются затраты на хранение продукции, стоимость денежных средств, а также сроки годности продукции: *StorageC(vl)* – стоимость хранения товарного запаса объемом *vl*, д.е./год; *MoneyC(vl)* – стоимость денежных средств, д.е./год; *DeficitC(vl)* – потери от дефицита, д.е./год (упущенный спрос); *ExpiredC(vl)* – потери от списания продукции с истекшим сроком годности, д.е./год.

Пользователь системы может ознакомиться с результатами расчета оптимального уровня сервиса на вкладке «Анализ» – «Уровень сервиса» (рис. 8).

Учет ограничений

При расчете заказа поставщику пользователь может задать ограничения сверху и снизу на такие параметры, как сумма заказа, число дней товарного запаса, объем заказа, вес заказа, число транспортных средств и их характеристики.

Чтобы иметь возможность выбора наиболее оптимального с точки зрения бизнес-логики пользователя товара, применяется механизм настраиваемых критериев важности товаров. В качестве кри-

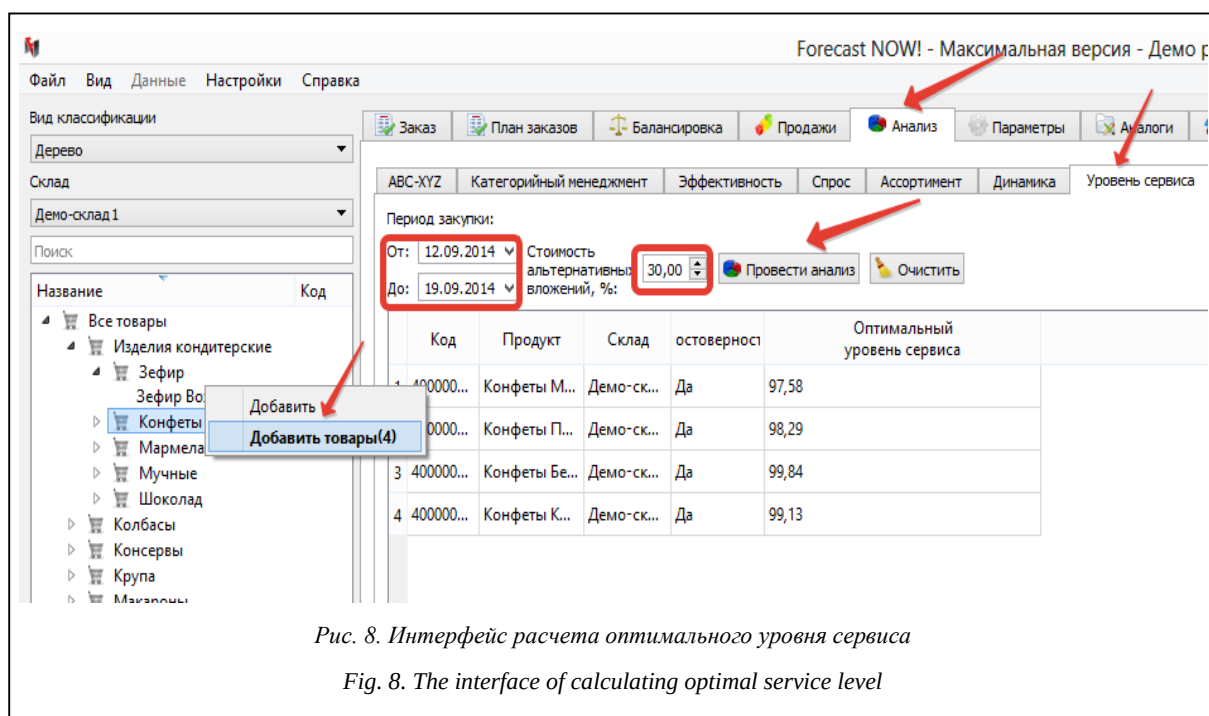


Рис. 8. Интерфейс расчета оптимального уровня сервиса

Fig. 8. The interface of calculating optimal service level

териев для выбора товаров доступны следующие: востребованность товара, плотность товара исходя из весогабаритных его характеристик, рентабельность товара, цена закупки товара, пользовательский критерий.

Аналитический блок

Аналитический блок позволяет оценивать эффективность управления товарными запасами как за конкретный временной промежуток, так и в динамике. Также данный блок предоставляет исходную информацию для категорийного менеджмента.

В аналитический блок входят следующие виды анализов: ABC-XYZ, «Эффективность», «Спрос», «Ассортимент», «Динамика», «Категорийный менеджмент» [15], «Обеспечение продаж», «Излишки».

Анализ ABC-XYZ позволяет проводить классический ABC-XYZ-анализ либо произвольный анализ, начиная от кросс-ABC-анализа, заканчивая ABC-XYZ-анализом по требуемым параметрам (выручка, количество, себестоимость, прибыль).

ABC-анализ позволяет оценить вклад того или иного товара по выбранной характеристике в общий объем по этой характеристике. Например, если проводится ABC-анализ по выручке, то в группу А попадают товары, приносящие наибольшую выручку, в группу В – среднюю, в С – наименьшую.

XYZ-анализ позволяет оценить степень вариативности (изменчивости) товаров по выбранному параметру: X – самая стабильная группа, Y – группа средней стабильности, Z – нестабильная.

Анализ «Эффективность» позволяет оценить, насколько эффективно в настоящий момент заказываются те или иные позиции и группы товаров. По товарным позициям или товарным группам, а также группировкам по любым пользовательским свойствам можно замерить ключевые показатели эффективности управления товарными запасами как в натуральном, так и в денежном выражении: оборачиваемость, уровень сервиса, упущенная прибыль, излишние запасы и т.п.

Анализ «Спрос» позволяет оценить изменение товарного запаса при изменении уровня сервиса, а также наглядно представить полученное распределение спроса на товарную позицию на заданный временной промежуток. Данная функция позволяет пользователю лучше понять механизм работы программы, а значит, больше доверять ее расчетам.

Анализ «Ассортимент» позволяет оценить ассортимент товаров предприятия на основе предопределенных бизнес-правил. Данные представляются в виде столбиковой диаграммы, разделенной на следующие группы:

- выбывший ассортимент – товары, которые в последний месяц (или другой период) не продавались и отсутствовали на складе;
- неликвид – товары, которые есть на складе, но в последний месяц (или другой период) по ним не было продаж;
- нет на складе – на сегодняшнюю дату этих товаров нет на складе;
- остатки не соответствуют продажам – группа предназначена для проверки того, насколько правильно реализован обмен данными или насколько в целом достоверны данные;

- отрицательные остатки;
- отсутствует закупочная или реализованная стоимость – ряд анализов по таким товарам будут некорректны, например, ABC-XYZ-анализ по прибыли или по стоимости;
- прерывистые продажи – группа товаров с прерывистым спросом, средний интервал между фактами продаж – больше 1,25 периода;
- продажи под заказ – группа товаров, привозимых поставщиком и продаваемых потребителям в один и тот же день.

Анализ «Динамика» позволяет анализировать эффективность работы склада в динамике, представленную в графическом виде. Результаты можно анализировать как по конкретному товару, так и по группе товаров. На одном графике можно сравнивать следующие показатели: средние продажи и остатки, уровень сервиса и оборачиваемость, прибыль и рентабельность запасов.

Анализ «Категорийный менеджмент» [15] позволяет выявлять ТОП-10 товарных групп по различным показателям эффективности и прочим аналитическим показателям, доступным в анализе «Эффективность». Он предназначен для управления ассортиментом и выявления проблемных товарных групп, поставщиков и т.п.

Анализ «Обеспечение продаж» позволяет оценить объем продаж, который может быть достигнут при выставленных параметрах системы, исходя из математического ожидания продаж и оптимального товарного запаса на месяц с заданным уровнем сервиса. Оптимальный товарный запас на месяц по сути является оценкой товарооборота за месяц.

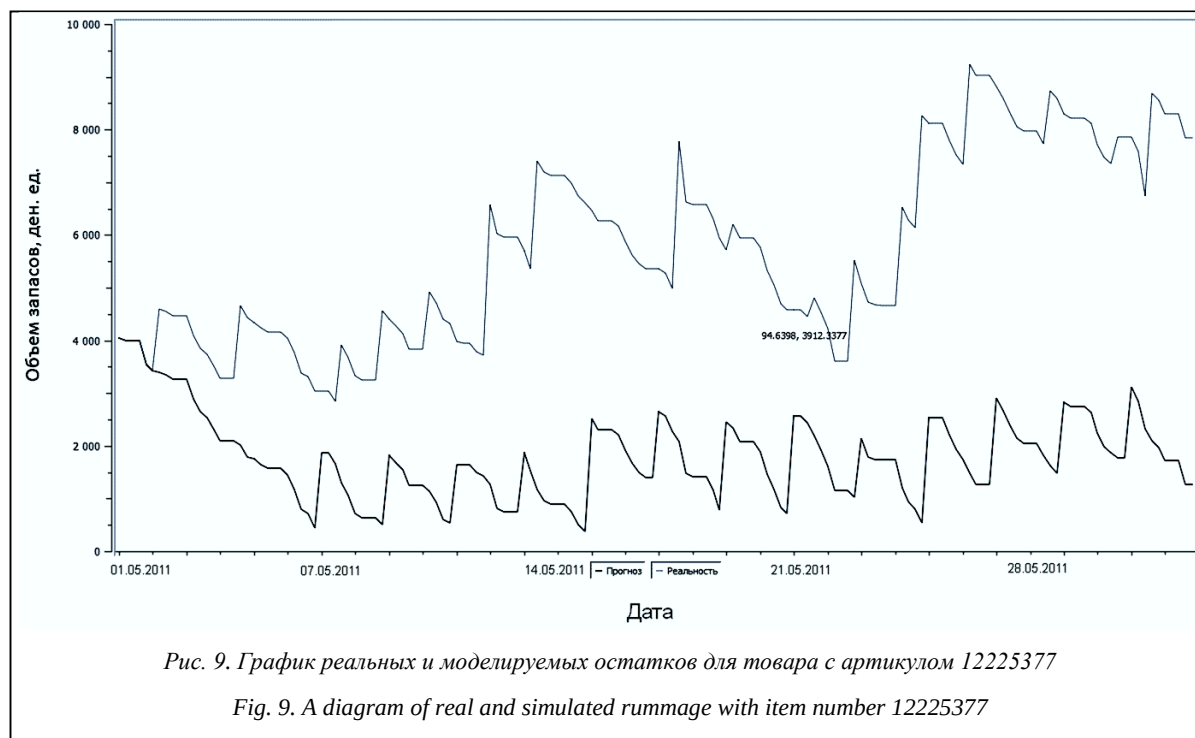
Анализ «Излишки» позволяет оценить динамику объема излишних запасов и объем запасов, не востребованный за разные периоды времени (год, месяц, неделя), в денежных единицах, а также увидеть понедельное распределение объема таких товаров.

Экспорт данных

Модуль экспорта данных позволяет сохранять результаты расчета заказа и проведенных анализов в форматах csv, xls, xlsx, html. Для сохранения в форматах xls и xlsx используется сторонняя библиотека LibXL (<http://libxl.com/>). Модуль обеспечивает обмен данными с КИС пользователя посредством обмена данными в формате csv.

Результаты внедрения системы

Опытное внедрение системы было осуществлено на одном из торговых предприятий города Твери. Для оценки экономической эффективности перед внедрением было проведено имитационное моделирование. При этом использовались данные предприятия о продажах и остатках за 2011 год. Моделировалось поведение склада и продаж при осуществлении закупок по рекомендациям программы в течение 2 месяцев. Закупки делались каждые две недели. Полученные результаты сопоставлялись с реальными остатками и продажами. При расчете экономии во внимание принимались возникшая экономия по закупочной цене, а также возможный дефицит в результате уменьшения запаса.



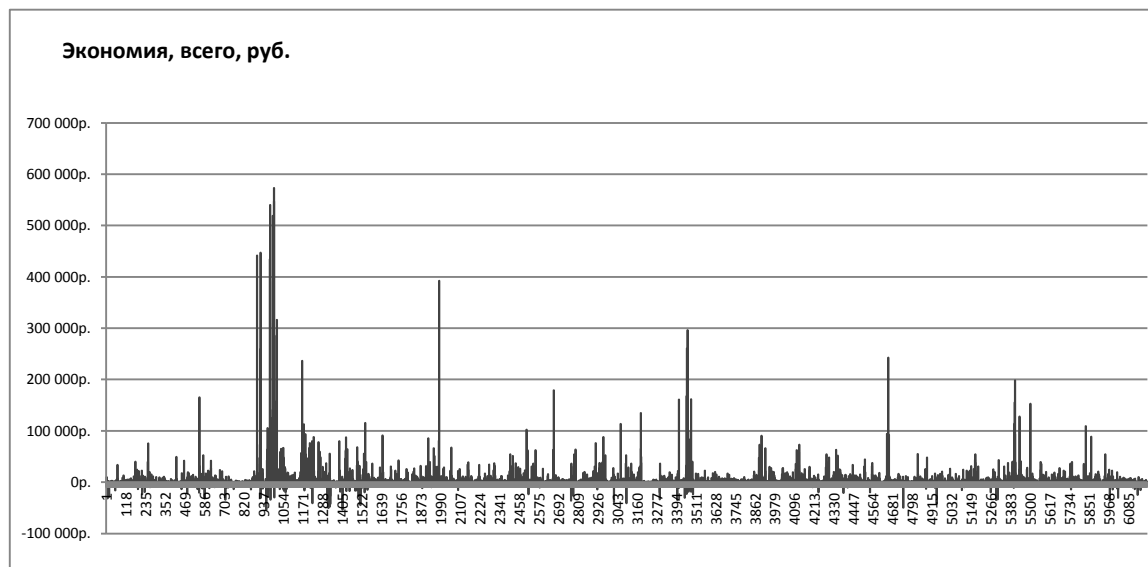


Рис. 10. Экономия в разрезе товарных позиций

Fig. 10. Economy in terms of goods items

В моделировании не учитывались товары, которые имели менее двух движений за период. Общая расчетная экономия в процентном соотношении составила 63 % от общего объема склада. Сравнение результатов моделирования с реальными историческими данными представлено на рисунке 9.

На рисунке 10 показана экономия в разрезе товарных позиций. По оси абсцисс – номер товарной позиции, по оси ординат – размер экономии по этой позиции. Стоит отметить, что не все денежные средства могут быть быстро высвобождены при использовании рекомендаций программы. Это связано с тем, что товары имеют разную степень ликвидности. Так, основная масса нерационально использованных оборотных средств сосредоточена на крайне медленно оборачивающихся позициях. Тем не менее, моделирование показало, что 15 % от оборотных средств могут быть высвобождены в течение 6 месяцев.

В настоящий момент программный продукт Forecast NOW! внедрен на пятнадцати торговых предприятиях России (Тверь, Москва, Владивосток, Новосибирск и др.), Казахстана (Петропавловск) и Беларуси (Минск). Общее число торговых точек, которые используют программный продукт в своей работе, превышает 100 структурных единиц. Экономический эффект составляет порядка 10–30 % дополнительной прибыли для предприятия за счет увеличения уровня сервиса на 7–15 %, сокращения уровня товарных запасов на 15–34 %, а также снижения числа неликвидов. Имеются положительные отзывы от клиентов по результатам работы с программным продуктом, а также от экспертов в области управления запасами и цепочками поставок. Программный продукт применяется в

учебном процессе университетов по направлению подготовки «Бизнес-информатика» [16]. Работа представлена [17] на различных профильных и научных конференциях.

Литература

1. Грицай А.А. и др. Программа оптимизации складских запасов Forecast NOW! Свид. о гос. регистр. программы для ЭВМ 2012613236.
2. Грицай А.А., Варанкин Д.А. Система управления товарными запасами Forecast NOW! Версия 3.0. Свид. о гос. регистр. программы для ЭВМ 2013619454.
3. Хайкин С. Нейронные сети. Полный курс. 2-е изд. М.: Вильямс, 2006. 114 с.
4. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы. 2-е изд. М.: Физматлит, 2006. 320 с.
5. Грицай А.А., Остренков Н.В. Forecast NOW! Балансировка складов. Свид. о гос. регистр. программы для ЭВМ 2015614336.
6. Грицай А.А. и др. Forecast NOW! Версия Автозапчасти. Свид. о гос. регистр. программы для ЭВМ 2015611669.
7. Dixon W.J. Analysis of Extreme Values. Ann. Math. Stat, 1950, vol. 21, no. 4, pp. 488–506.
8. Dixon W.J. Ratios Involving Extreme Values. Ann. Math. Stat, 1951, vol. 22, no. 1, pp. 68–78.
9. Большев Л.Н., Убайдуллаева М. Критерий Шовенэ в классической теории ошибок // Теория вероятностей и ее применение. 1974. Т. XIX. № 4. С. 714–723.
10. Tietjen G.L., Moore R.H. Some Grubbs-Type Statistics for the De-tecton of Several Outliers. Technometrics, 1972, vol. 14, no. 3, pp. 583–597.
11. Пагурова В.И. О критерии Шовенэ для обнаружения нескольких выбросов // Теория вероятностей и ее применение. 1985. Т. 30. № 3. С. 558–561.
12. Грицай А.А. Расчет оптимального запаса товаров частого спроса с использованием нейронных сетей // Математика, статистика и информационные технологии в экономике, управлении и образовании: сб. тр. III Междунар. науч.-практич. конф. Тверь: Изд-во Тверского гос. ун-та, 2014.
13. Грицай А.А., Варанкин Д.А. Forecast NOW! Мульти-этапное планирование. Свид. о гос. регистр. программы для ЭВМ 2015614338.

14. Грицай А.А. Определение эффективной структуры входных данных для обучения нейронной сети решению задачи прогнозирования спроса // Вестн. Тверского гос. ун-та: Сер. Прикладная математика. 2014. № 2. С. 123–134.

15. Грицай А.А., Остренков Н.В. Forecast NOW! Категорийный менеджмент. Свид. о гос. регистр. программы для ЭВМ 2015614337.

16. Грицай А.А., Романова Е.Ю. Интегративное значение изучения информационных систем в профессионально-матема-

тической подготовке по специальности «Бизнес-информатика» (на примере системы Forecast NOW!) // Информационные технологии в науке и образовании: сб. тр. Всерос. науч.-практич. конф. Чебоксары: Изд-во Чувашского гос. пед. ун-та, 2013. С. 28–31.

17. Викторов Е.А., Грицай А.А. Тверские Интернет-технологии // Система управления товарными запасами Forecast NOW! Технология и коммерциализация: сб. трудов науч.-практич. конф. Тверь: Изд-во Тверского гос. ун-та, 2013. С. 38–42.

DOI: 10.15827/0236-235X.112.219-230

Received 01.06.15

INTELLIGENT INFORMATION SYSTEM FOR RETAIL DEMAND FORECASTING AND INVENTORY OPTIMIZATION

Gritsay A.A., Postgraduate Student, allgrit@forecastnow.ru

(Tver State University, Zhelyabova St. 33, Tver, 170100, Russian Federation)

Abstract. The paper presents inventory management and stock optimization system for retail trade. The system is focused on small and medium retailers and wholesale trade customers. The estimated savings make a total of 20–30 % of the stock. System can increase satisfied demand by 5–7 %. The article describes the main features of the system: data preprocessing, demand forecasting, calculation of the optimal service level, calculation of optimal inventory, order calculation for a supplier. Data preprocessing cleans data from sharp emissions, marketing campaigns impact, holiday impact and the lack of goods in the warehouse. Demand forecasting is based on the preprocessed time series of the sales. The characteristics of sales history time series are calculated for each product. The prediction algorithm is selected according to the characteristics. Given service level defines the reserve stock. The optimal service level is calculated according to the strategy of trading company investment return maximization. Optimal inventory is calculated based on the calculated service level and the chosen forecasting algorithm. Available goods in stock, goods in transit, as well as provisions for individual customers determine the optimal order size of the item.

After that row order restrictions are applied (e.g. packaging multiplicity), and then order restrictions to a supplier as a whole (e.g. weight and dimensional). The paper provides an example of applying the system and the conclusion about economic effect from its implementation in a real trading enterprise.

Keywords: inventory management; inventory stock optimization; inventory management system; neural networks; demand forecasting; optimal service level; order to a supplier.

References

1. Gritsay A.A. *Programma optimizatsii skladskikh zapasov Forecast NOW!* [Forecast NOW! Software for stock resources optimization]. 2012. Software State Registration Certificate no. 2012613236.
2. Gritsay A.A., Varankin D.A. *Sistema upravleniya tovarnymi zapasami Forecast NOW! Versiya 3.0* [Stock Goods Management System Forecast NOW! Version 3.0]. 2013. Software State Registration Certificate no. 2013619454.
3. Khaykin S. *Neyronnye seti. Polny kurs* [Neural Networks. Full Course]. 2nd ed. Moscow, Vilyams Publ., 2006, 114 p.
4. Gladkov L.A., Kureychik V.V., Kureychik V.M. *Geneticheskie algoritmy* [Genetic algorithms]. 2nd ed. Moscow, Fizmatlit Publ., 2006, 320 p.
5. Gritsay A.A., Ostrenkov N.V. *Forecast NOW! Balansirovka skladov* [Forecast NOW! Storage Balancing]. 2015. Software State Registration Certificate no. 2015614336.
6. Gritsay A.A. *Forecast NOW! Versiya Avtozapchasti* [Forecast NOW! Version Auto Parts]. 2015. Software State Registration Certificate no. 2015611669.
7. Dixon W.J. Analysis of Extreme Values. *Ann. Math. Stat.* 1950, vol. 21, no. 4, pp. 488–506.
8. Dixon W.J. Ratios Involving Extreme Values. *Ann. Math. Stat.* 1951, vol. 22, no. 1, pp. 68–78.
9. Bolshev L.N., Ubaydullaeva M. Shovene criteria in classical error theory oshibok. *Teoriya veroyatnostey i ee primeneniye* [The Theory of Probability and its application], 1974, vol. XIX, no. 4, pp. 714–723 (in Russ.).
10. Tietjen G.L., Moore R.H. Some Grubbs-Type Statistics for the Detection of Several Outliers. *Technometrics*. 1972, vol. 14, no. 3, pp. 583–597.
11. Pagurova V.I. On Shovene criteria to detect several outliers. *Teoriya veroyatnostey i ee primeneniye* [The Theory of Probability and its application]. 1985, vol. 30, no. 3, pp. 558–561 (in Russ.).
12. Gritsay A.A. Raschet optimal'nogo zapasa tovarov chastogo sprosa s ispol'zovaniem neyronnykh setey. *Matematika, statistika i informatsionnye tekhnologii v ekonomike, upravlenii i obrazovanii: sb. tr. III Mezhdunar. nauch.-prakt. konf.* [Proc. of the 3rd Int.]. Tver, 2014 (in Russ.).
13. Gritsay A.A., Varankin D.A. *Forecast NOW! Multieshelonnoe planirovanie* [Forecast NOW! Multi-echelon Planing]. 2015. Software State Registration Certificate no. 2015614338.
14. Gritsay A.A. Input data effective structure determination for training neural network to solve demand forecast problem. *Vestnik Tverskogo gos. universiteta: Seriya "Prikladnaya matematika"* [Herald of Tver State Univ. Series: Applied Mathematics]. 2014, no. 2, pp. 123–134.
15. Gritsay A.A., Ostrenkov N.V. *Forecast NOW! Kategoriyny menedzhment* [Forecast NOW! Categorical Management]. 2015. Software State Registration Certificate no. 2015614337.
16. Gritsay A.A., Romanova E.Yu. Integrative meaning of information systems research in professional mathematical foundation majoring in Business Informatics (in case of Forecast NOW!). *Vseross. nauch.-praktich. konf. "Informatsionnye tekhnologii v nauke i obrazovanii": sb. tr.* [Proc. of the All-Russian Science and Practice Conf. "IT in Science and Education"]. Cheboksary, 2013.
17. Viktorov E.A., Gritsay A.A. Tver Internet technologies. *Sb. tr. nauchno-praktich. konf. "Sistema upravleniya tovarnymi zapasami Forecast NOW! Tekhnologiya i kommersializatsiya"* [Proc. of the Science and Practice Conf. "Stock Goods Management System Forecast NOW! Technology and Commoditization"]. Tver, 2013.

УДК 629.4.067

Дата подачи статьи: 23.06.15

DOI: 10.15827/0236-235X.112.231-236

ИНТЕЛЛЕКТУАЛЬНАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ РЕШЕНИЯ ЗАДАЧ ПРОГНОЗИРОВАНИЯ НЕИСПРАВНОСТЕЙ ВАГОННОГО ОБОРУДОВАНИЯ НА ЖЕЛЕЗНОДОРОЖНОМ ТРАНСПОРТЕ

(Работа выполнена при поддержке грантов №№ 12-07-13117 офи_м_РЖД, 13-07-13160 офи_м_РЖД)

Е.И. Иванова, аспирант, kateivanova@gmail.com

(Тверской государственный университет, ул. Желябова, 33, г. Тверь, 170100, Россия)

Повышение уровня безопасности функционирования железнодорожного транспорта представлено в «Стратегии развития железнодорожного машиностроения до 2030 года» как одна из важнейших проблем, актуальных для научных исследований. В числе первоочередных задач, поставленных ОАО «РЖД» в данной стратегии, – создание комплексных систем диагностики и прогнозирования технического состояния инфраструктуры подвижного состава. Анализ публикаций на данную тему показывает, что в них нет информации об использовании подобных систем и средств на подвижных железнодорожных составах, представленных на российском рынке вагоностроения.

В статье описан один из подходов к решению упомянутой проблемы – применение интеллектуальной информационной системы, ориентированной на решение задач прогнозирования неисправностей вагонного оборудования на железнодорожном транспорте. Отражены основные научные результаты разработки данной системы, описаны архитектура системы и ее особенности, используемые в процессе реализации алгоритмы и средства реализации, результаты и анализ применения системы на реальных данных парка подвижного состава Федеральной пассажирской компании.

Описанная система во многих смыслах является адаптируемой и универсальной. Она ориентирована на применение в качестве системы прогнозирования неисправностей пассажирских вагонов, но в дальнейшем может быть адаптирована и применена к другим элементам железнодорожного транспорта (к грузовым вагонам, локомотивам, электропоездам и так далее). Ее подсистемы не зависят друг от друга и дают возможность экспериментировать с применением различных алгоритмов в рамках ее подсистем, не затрагивая работу системы в целом.

Данная система реализована при финансовой поддержке РФФИ в рамках конкурса ориентированных фундаментальных исследований по актуальным междисциплинарным темам в интересах РЖД 2013 года.

Работа описанной системы проверена на ресурсах вагонов производства Тверского вагоностроительного завода, оснащенных новейшей системой контроля, диагностики и управления парка подвижного состава Федеральной пассажирской компании.

Ключевые слова: информационная система, железнодорожный транспорт, диагностика, прогнозирование, принятие решений, мягкие вычисления, генетический алгоритм, нейронная сеть, нейронная эхо-сеть.

Эффективное функционирование железнодорожного транспорта Российской Федерации играет исключительную роль в создании условий для модернизации, перехода на инновационный путь развития и устойчивого роста национальной экономики, способствует созданию условий для обеспечения лидерства России в мировой экономической системе. Повышение уровня безопасности функционирования железнодорожного транспорта выделено в «Стратегии развития железнодорожного машиностроения до 2030 года» как одно из важнейших направлений научных исследований.

Технический и технологический уровень железнодорожного подвижного состава довольно низкий, оборудование, установленное на железнодорожных подвижных составах, часто выходит из строя. Как следствие, тратятся огромные суммы на обслуживание и ремонт подвижных составов.

В процессе эксплуатации вагонов часто возникают аварийные ситуации, обусловленные износом ресурсов парка подвижного состава, а также неисправностью установленного на подвижных составах оборудования. Поэтому одной из важнейших задач, поставленных ОАО «РЖД» до 2030 года, является создание комплексных систем диагностики и прогнозирования технического состояния инфраструктуры подвижного состава.

В публикациях на данную тему нет упоминаний об использовании подобных систем и средств на подвижных железнодорожных составах, представленных на российском рынке вагоностроения. Проблема обеспечения безопасности затрагивается лишь в области автостроения [1]. Она частично решается путем сведения ее к задаче автоматического выявления дефектов, а потому является актуальной и новой.

В качестве одного из подходов к решению проблемы в данной статье предлагается применение интеллектуальной информационной системы, ориентированной на решение задач прогнозирования неисправностей вагонного оборудования на железнодорожном транспорте.

Особенностями данной системы являются ее архитектура, допускающая декомпозицию на более узконаправленные независимые подсистемы, и использование методов эволюционного программирования, алгоритма машинного обучения и особой разновидности нейронной сети.

Архитектура интеллектуальной информационной системы и ее функционал

Представленная информационная система является распределенной, ее подсистемы выполняют

разные функции и работают независимо друг от друга. Она включает в себя подсистемы сбора данных, предварительной обработки данных, оптимизации данных, прогнозирования.

В подсистеме сбора данных выполняется сбор данных с вагонного оборудования. Сбор производится локально на каждом вагоне, затем данные со всех вагонов собираются и сводятся в централизованную БД. В подсистеме формирования срезов из БД извлекаются необходимые данные и затем специальным образом обрабатываются. В подсистеме оптимизации на основе специальных алгоритмов данные обрабатываются и оптимизируются за счет сокращения лишних, не влияющих на дальнейшее прогнозирование данных. В подсистеме прогнозирования производится построение прогноза наличия или отсутствия конкретной неисправности.

Подсистемы имеют входы и выходы, причем выходные данные одних подсистем являются входными для других. Преимущества данной архитектуры в легкости изменения модулей, возможности применять и сравнивать различные алгоритмы, меняя подсистемы, но не изменяя концепцию системы в целом.

Подсистема сбора данных. В системе контроля, диагностики и управления (СКДУ) пассажирских вагонов [2] реализована подсистема сбора данных с вагонов, состоящая из модуля отслеживания изменений параметров устройств СКДУ вагона. Модуль реализован на языке python и запускается вместе с СКДУ вагона. В процессе работы СКДУ вагона он следит за параметрами и в случае их изменения фиксирует значения с системными временными метками.

Этот модуль снимает данные о возникающих неисправностях оборудования, зафиксированных СКДУ вагона, с системными временными метками возникновения.

Преимущество описанного способа сбора данных в том, что изменения параметров устройств вагона фиксируются по факту: невозможна ситуация, когда параметр имеет не зафиксированное в БД промежуточное значение, как в случае периодического снятия данных.

Сбор данных производится на различных уровнях данных подвижного состава:

- уровень устройства или системы устройств вагона;
- уровень вагона – совокупность устройств или систем устройств;
- уровень подвижного состава – совокупность вагонов.

Сбор данных производится как локально на каждом вагоне, так и на штабном вагоне с помощью СКДУ состава, владеющей основной информацией по всем вагонам состава. Данные накапливаются и хранятся в СКДУ вагона и СКДУ состава, а затем передаются и сводятся в централизованную БД под управлением СУБД SQLite [3, 4].

Способы передачи данных с вагонов могут быть разными: мобильная связь, физический носитель, спутниковая связь, Интернет. Каждый способ имеет свои преимущества и недостатки, в основном связанные со скоростью передачи данных, требованиями к обслуживанию и стоимостью реализации и сопровождения.

В СКДУ реализована передача данных с помощью физического носителя, предполагающая периодическое выполнение сервисными работниками процедуры копирования данных с СКДУ вагона на внешний накопитель, а затем физическую доставку данных в подсистему к серверу с БД и последующее наполнение БД.

Данный способ сравнительно недорогой, но не оперативный.

Подсистема предварительной обработки данных. После наполнения БД информацией, полученной с вагонов, производится извлечение данных, связанных с конкретным объектом (устройством, системой, составом), и их обработка для последующего анализа.

БД удобна для структурированного хранения данных. Для более эффективной обработки данных с конкретных устройств используется срез данных, под которым понимается следующий набор данных: $t_i, V_{i1}, \dots, V_{ip}$, где t_i – i -й момент времени снятия среза данных в формате «ДД.ММ.ГГГГ ЧЧ:ММ:СС»; $V_{i1}, \dots, V_{ip}$ – значения переменных и неисправностей $N_1, \dots, N_p$ в данный момент времени, причем переменные и неисправности могут быть из разных уровней данных подвижного состава. На рисунке 1 представлена схема срезов данных – значений переменных и неисправностей.

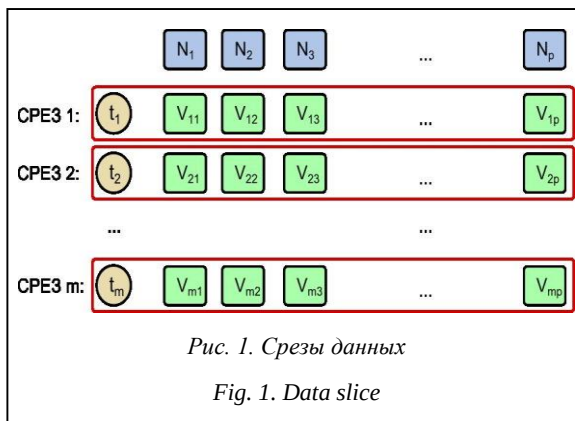


Рис. 1. Срезы данных

Fig. 1. Data slice

В момент времени t_i переменная или неисправность с именем N_j будет иметь значение V_{ij} при $i \in \{1, \dots, m\}, j \in \{1, \dots, p\}$.

В подсистеме формирования срезов данных производится извлечение данных и формирование множества срезов данных с шагом в 5 секунд с помощью программных модулей, реализованных на языках Ruby и R [5].

Подсистема оптимизации данных. Следующий этап обработки – выявление зависимостей па-

параметров и неисправностей между собой и фильтрация параметров, не влияющих на конкретные неисправности. Это задача подсистемы оптимизации данных. В данную подсистему входит модуль, основанный на алгоритме построения случайного леса [6], который заключается в использовании ансамбля деревьев принятия решений. Алгоритм сочетает в себе две основные идеи: метод бэггинга Бреймана и метод случайных подпространств Тин Кам Хо [7].

Разбиение узла в случайном лесе осуществляется на основании данных об оптимальном разбиении для данного узла случайно выбранного подмножества переменных (параметров и неисправностей) из срезов. Каждое дерево строится до полного исчерпания подмножества.

Алгоритм рассчитан на эффективную обработку больших объемов данных с большим количеством переменных и, что важно в рамках этой задачи, предполагает методы оценивания значимости отдельных переменных в построенной модели.

В программном модуле учитываются следующие критерии значимости переменных при построении случайного леса:

- среднее снижение точности (%IncMSE), определяемое с помощью перестановки значений переменных тестового множества и сравнения прогноза значений переменных с прогнозом исходного множества, нормированного по стандартной ошибке;

- среднее снижение коэффициента Gini (IncNodePurity), отображающее качество NodePurity с помощью коэффициента Gini разделения дерева по переменным; от родителей к потомкам значение коэффициента Gini уменьшается, значимость переменной определяется суммой сокращений коэффициента Gini по всем деревьям.

Критерии %IncMSE и IncNodePurity пропорциональны значимости переменной.

Достоинством данного алгоритма является простая настройка – необходимо указать количество случайных узлов в подмножестве для каждого узла и количество деревьев принятия решений в случайном лесе.

Как и любой другой, алгоритм имеет недостатки: склонность к переобучению на сильно зашумленных задачах и порой большой размер памяти (требуется $O(NK)$ для хранения модели, где K – число деревьев принятия решений, а N – количество примеров в обучающей выборке).

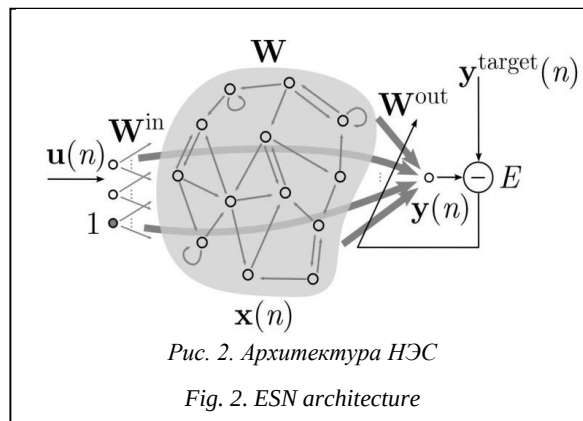
Программный модуль реализован на языке R с использованием пакетов `rpart` – для построения дерева принятия решений, `randomForest` – для построения случайного леса.

С помощью алгоритма построения случайного леса выявляются параметры с наибольшей важностью для исследуемой неисправности и выполняется последующая фильтрация исходных данных.

Подсистема прогнозирования. Состоит из модуля прогнозирования неисправностей, основанного на разновидности рекурсивных нейронных сетей, называемой *нейронной эхо-сетью* (НЭС) [8].

Архитектура НЭС (рис. 2) включает три слоя:

- входной слой, на который подаются входные массивы данных;
- скрытый слой, охваченный обратными связями и называемый также динамическим резервуаром;
- выходной слой, считывающий информацию из скрытого слоя.



На рисунке 2 $u(n) \in R^{N_u}$ – входной сигнал; $y^{target}(n) \in R^{N_y}$ – выходной целевой сигнал; $n = 1, \dots, T$ – дискретное время; T – число элементов в обучающем множестве; $y(n) \in R^{N_y}$ – получаемый сигнал; $x(n) \in R^{N_x}$ – вектор активаций нейронов резервуара; $W^{in} \in R^{N_x \times (1+N_u)}$, $W \in R^{N_x \times N_x}$, $W^{out} \in R^{N_y \times (1+N_x)}$ – матрицы входных весов, резервуара и выходных весов соответственно.

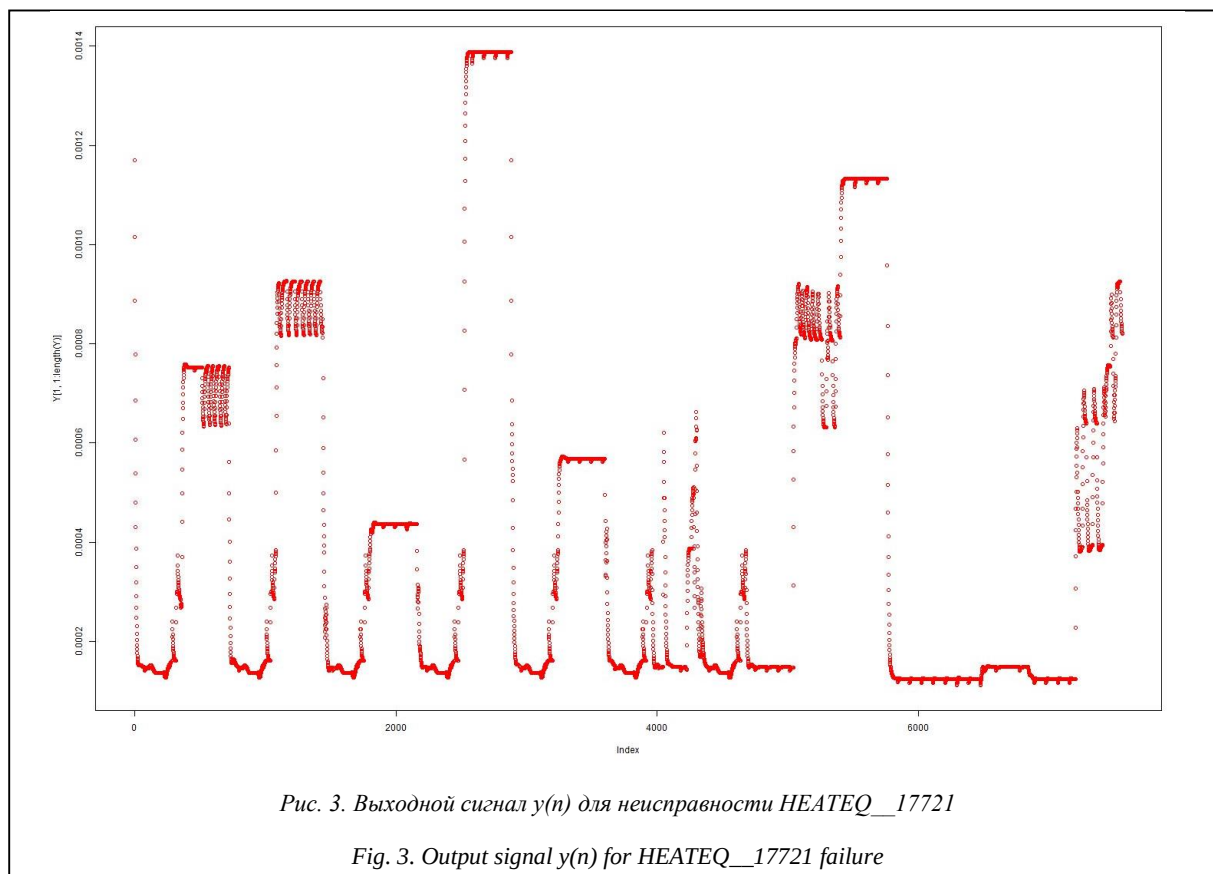
Мерой ошибки служит среднеквадратичная ошибка: $E(y, y^{target}) = \frac{1}{T} \sum_{n=1}^T (y(n) - y^{target}(n))^2$.

Для оптимизации обучения НЭС был применен генетический алгоритм, позволяющий наилучшим образом обучить НЭС, а также получить оптимальный экземпляр [9].

Работа информационной системы на реальных данных

Представленная интеллектуальная информационная система внедрена и испытана при производстве двухэтажных вагонов Тверским вагоностроительным заводом. Для анализа было взято устройство КУЭВ, входящее в состав СКДУ вагона [10]. Параметры КУЭВ являются следующими структурами данных:

- 1) $\{V, S\}$, где $V \in \{ON, OFF\}$, а $S \in \{NORMAL, KZ, NA, FAULT\}$;
- 2) $\{N, S\}$, где $N \in R$, а $S \in \{NORMAL, KZ, NA, FAULT\}$.



Взятый образец КУЭВ содержит данные по 173 переменным. Из БД, содержащей данные за несколько дней, были взяты данные за 3 часа, сформировано множество срезов с периодом в 5 секунд. В качестве прогнозируемой выбрана неисправность устройства HEATEQ с кодом 17721 (далее – HEATEQ_17721). Период предсказания – 15 минут. Данное значение выбрано для достижения компромисса между скоростью работы нейронной сети и качеством предсказания.

Для массива срезов был построен случайный лес, выявлены наиболее значимые переменные и отфильтрованы исходные данные:

```
> IMP.global
[1] "CONFIG_7016"          "HEATEQ_17721"
[3] "HEATEQ_17004"         "HEATEQ_17003"
[5] "HEATEQ_17002"         "OUT_UKV_PCH_PROTECT_RST"
[7] "IN_BOILER_COVER_OPENED" "IN_SIGNAL_LIGHTS"
[9] "IN_GANGWAY"           "IN_LOW_HEAT_CARRIER_LEVEL"
[11] "OUT_SIGNAL_LIGHTS"    "OUT_EM_VALVE_ON"
[13] "HEATEQ_17005"         "CONFIG_7017"
[15] "OUT_2_2_PCH"          "IN_UKV_SUPPLY_FAN"
[17] "OUT_UKV_COMPR_ON"     "OUT_2_1_PCH"
```

Затем была построена НЭС с уровнем утечки $\alpha = 1$.

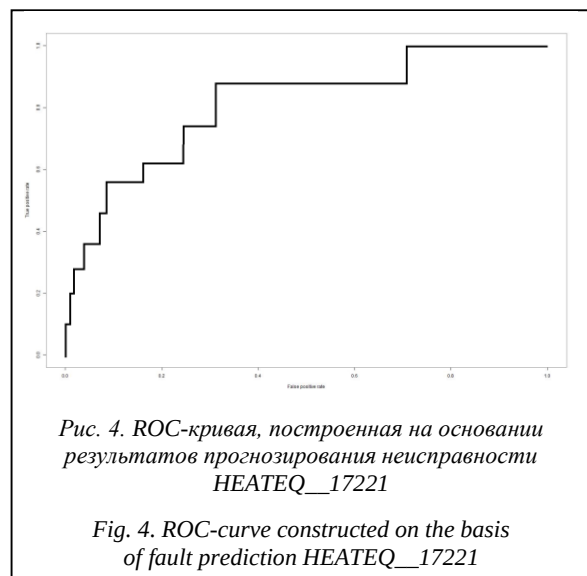
Для проверки на вход НЭС подавались срезы из контрольной выборки, а затем вычислялось значение выходного вектора. Значения, полученные на выходе, сокращались и обрабатывались с помощью пороговой функции.

Выходной сигнал $y(n)$ для неисправности HEATEQ_17721 представлен на рисунке 3 [11].

Значение среднеквадратичной ошибки равно 0,238.

Правильный прогноз был построен с точностью примерно 75 %, из которых примерно 78 % в случае возникновения неисправности и примерно 70 % в случае отсутствия неисправности.

Для оценки результатов работы алгоритма использовался метод ROC-кривой [12]. Он выражается в виде характеристики качества бинарного классификатора, а также зависимости доли верных положительных классификаций от доли ложных положительных классификаций при изменении порога решающего правила.



На основании результатов прогнозирования была построена ROC-кривая, представленная на рисунке 4.

Площадь под ROC-кривой называется интервалом AUC (Area Under Curve) – агрегированная характеристика качества классификации, не зависящая от соотношения цен ошибок [13]. Чем больше значение интервала AUC, тем лучше модель классификации в соответствии с таблицей.

Классификация качества модели по значению интервала AUC

Model quality classification by AUC interval value

Интервал AUC	Качество модели
0,9–1,0	Отличное
0,8–0,9	Очень хорошее
0,7–0,8	Хорошее
0,6–0,7	Среднее
0,5–0,6	Неудовлетворительное

В данном случае значение $AUC = 0,811942$, что свидетельствует о хорошем качестве построенной модели.

В заключение отметим, что в статье представлена интеллектуальная информационная система, ориентированная на решение задач прогнозирования неисправностей вагонного оборудования на железнодорожном транспорте, продемонстрированы и проанализированы ее возможности при работе на реальных данных.

Система во многих смыслах является адаптируемой и универсальной. Она ориентирована на применение на пассажирских вагонах, но в дальнейшем может быть применена и к другим элементам железнодорожного транспорта (к грузовым вагонам, локомотивам, электропоездам и так далее). Ее подсистемы не зависят друг от друга и дают возможность экспериментировать с применением различных алгоритмов в рамках подсистем, не затрагивая работу системы в целом.

Данная система реализована с использованием ресурсов Тверского вагоностроительного завода, оснащенных новейшей системой СКДУ вагонов и составов парка подвижного состава АО «ФПК», и рекомендуется для внедрения в производство и эксплуатацию вагонов в качестве прогнозирующей компоненты СКДУ. Внедрение интеллектуальной информационной системы позволит одновременно снизить расходы на обслуживание железнодорожного транспорта и увеличить его безопасность. Раннее прогнозирование неисправностей дает возможность предотвратить их или значительно снизить последствия их влияния на другие элементы как вагона, так и состава, а также сократить сроки и стоимость технического обслуживания железнодорожных составов.

В качестве развития данной системы предполагается составление статистики возникновения неисправностей и поведения вагонного оборудования в зависимости от них, что позволит выявить недостатки в технической и алгоритмической частях реализации оборудования, а также составить рейтинги качества поставщиков оборудования.

Литература

1. Ford Classification Challenge. 2007. URL: http://home.comcast.net/~nn_classification/ (дата обращения: 20.06.2015).
2. Иванова Е.И., Гордеев Р.Н., Михайлов В.В., Северов А.В., Язенин А.В. Модель централизованной интеллектуальной информационной системы для решения задач диагностики и прогнозирования неисправностей вагонного оборудования и управления им на железнодорожном транспорте // Нечеткие системы и мягкие вычисления. 2012. Т. 7. № 2. С. 51–71.
3. Гришина Е.Н., Иванова Е.И., Михайлов В.В., Северов А.В., Солдатенко И.С., Судаков А.Г., Ханьгин М.А., Сорокин С.В., Язенин А.В. Интеллектуальная информационная система единого центра обработки данных ОАО «РЖД» для решения задач диагностики и прогнозирования нештатных ситуаций поведения вагонного оборудования подвижных составов на железнодорожном транспорте // Интеллектуальные системы на транспорте: сб. матер. IV Междунар. науч.-практич. конф. (С.-Петербург, 3–4 апреля 2014). СПб: Лик, 2014. С. 93–100.
4. Гришина Е.Н., Иванова Е.И., Михайлов В.В., Северов А.В., Солдатенко И.С., Судаков А.Г., Ханьгин М.А., Сорокин С.В., Язенин А.В. Модель интеллектуальной информационной системы прогнозирования неисправностей вагонного электрооборудования на железнодорожном транспорте // Интеллектуальные системы управления на железнодорожном транспорте ИСУЖТ-2013: тр. II науч.-технич. конф. М.: Изд-во НИИАС. С. 184–187.
5. Гришина Е.Н., Рыжова М.Н. Программа предварительной обработки и визуализации данных о состоянии оборудования железнодорожных вагонов // Свид. о гос. регистр. прогр. для ЭВМ № 2014615052.
6. Random Forest. URL: http://ru.wikipedia.org/wiki/Random_forest (дата обращения: 20.06.2015).
7. Чистяков С.П. Случайные леса: обзор // Тр. Карельского науч. центра РАН. 2013. № 1. С. 117–136.
8. Иванова Е.И., Сорокин С.В. Использование искусственных нейронных эхо-сетей в системе прогнозирования неисправностей вагонного электрооборудования на железнодорожном транспорте // Нечеткие системы и мягкие вычисления. 2013. Т. 8. № 1. С. 45–57.
9. Иванова Е.И., Сорокин С.В. Оптимизатор нейросетевых моделей прогнозирования неисправностей оборудования железнодорожных вагонов // Свид. о гос. регистр. прогр. для ЭВМ № 2014615050.
10. Сорокин С.В., Сорокина И.В., Язенин А.В. Прогнозирование неисправностей оборудования с использованием нейронных сетей и нечеткого вывода // Техника железных дорог. 2014. № 2 (26). С. 64–74.
11. Иванова Е.И. Методы прогнозирования неисправностей вагонного оборудования пассажирских железнодорожных подвижных составов на основе многослойной нейронной сети с использованием генетического алгоритма, а также нейронных эхо-сетей: сравнительный анализ // Вестн. ТвГУ. Сер. Прикладная математика. 2014. № 2. С. 67–79.
12. Plotting and Interpreting an ROC Curve. URL: <http://gim.unmc.edu/dxtests/roc2.htm> (дата обращения: 20.06.2015).
13. The Area Under a ROC Curve. URL: <http://gim.unmc.edu/dxtests/roc3.htm> (дата обращения: 20.06.2015).

DOI: 10.15827/0236-235X.112.231-236

Received 23.06.15

**INTELLIGENT INFORMATION SYSTEM
FOR ROLLING STOCK CARRIAGE EQUIPMENT FAILURES FORECASTING ON RAILWAYS**
(The work has been done with support of grants no. 12-07-13117 *офу_м_РЖД*, 13-07-13160 *офу_м_РЖД*)

Ivanova E.I., Postgraduate Student, kkateivanova@gmail.com
(Tver State University, Zhelyabov St. 33, Tver, 170100, Russian Federation)

Abstract. The “Strategy of railway engineering development until 2030” presents rail transport increased safety as one of the most important problems for research. One of the main goals of JSC “Russian Railways” in this strategy is creation of complex systems for diagnostics and forecasting of rolling stock infrastructure technical condition. Analysis of publications on the subject shows that there is no information on using such systems on rolling stock on the Russian market.

The article describes one approach to solve the abovementioned problem, which is using an intelligent information system focused on solving problems of equipment fault forecasting in rail transport. The article describes the main scientific results of such system development, the system architecture and its features, algorithms and implementation means, results and analysis of using this system on real data of rolling stock cars of JSC “FIC”.

The described system is an adaptable and versatile in many ways. It is applied as a forecasting system to passenger cars, but in the future it can be adjusted and applied to other rail transport elements (freight cars, locomotives, electric, etc.). Its subsystems are independent and it is possible to try different algorithms within its subsystems without affecting the system as a whole.

This system is realized with support from Russian Foundation for Basic Research within the competition of oriented basic research on important cross-disciplinary subjects on behalf of JSC “Russian Railways”.

The developed system is tested on the car resources of JSC “TCW” equipped with the latest SKDU system of JSC “FIC” rolling stocks.

Keywords: information system, rail transport, diagnostics, forecast, decision making, soft computing, genetic algorithm, neural network, echo state network.

References

1. *Ford Classification Challenge*. 2007. Available at: http://home.comcast.net/~nn_classification/ (accessed June 20, 2015).
2. Ivanova E.I., Gordeev R.N., Mikhaylov V.V., Severov A.V., Yazenin A.V. Model of centralized intelligent information system for diagnosis and prediction of failure of rolling equipment on railways. *Nechetkie sistemy i myagkie vychisleniya* [Fuzzy Systems and Soft Computing]. 2012, vol. 7, no. 2, pp. 51–71 (in Russ.).
3. Grishina E.N., Ivanova E.I., Mikhaylov V.V., Severov A.V., Soldatenko I.S., Sudakov A.G., Khanygin M.A., Sorokin S.V., Yazenin A.V. JSC “Russian Railways” intelligent information system unified data processing center for solving problems of diagnosis and prediction of failure of rolling equipment on railways. *Intellektualnye sistemy na transporte: sb. mater. IV Mezhdunar. nauch.-praktich. konf.* [Proc. of the 4th Int. Science and Practice Conf. “Transport Intelligent Systems”]. St.-Petersburg, Lik Publ., 2014, pp. 93–100 (in Russ.).
4. Grishina E.N., Ivanova E.I., Mikhaylov V.V., Severov A.V., Soldatenko I.S., Sudakov A.G., Khanygin M.A., Sorokin S.V., Yazenin A.V. A model of centralized intelligent information system for prediction of failure of rolling equipment on railways. *Intellektualnye sistemy upravleniya na zheleznodorozhnom transporte ISUZhT-2013: tr. II nauch.-tekhnich. konf.* [Proc. of the 2nd Science and Practice Conf. “Railways Intelligent Management Systems” (ISUZhT-2013)]. Moscow, NIIAS Publ., pp. 184–187 (in Russ.).
5. Grishina E.N., Ryzhova M.N. *Programma predvaritel'noy obrabotki i vizualizatsii dannykh o sostoyanii oborudovaniya zheleznodorozhnykh vagonov* [railway car equipment data preprocessing and visualization software]. PC program State Registr. Certif. no. 2014615052 (in Russ.).
6. *Random Forest*. Available at: http://ru.wikipedia.org/wiki/Random_forest (accessed June 20, 2015).
7. Chistyakov S.P. Random forests: a review. *Tr. Karelskogo nauch. tsentra RAN* [Transactions of the Karelian Research Centre of RAS]. 2013, no. 1, pp. 117–136 (in Russ.).
8. Ivanova E.I., Sorokin S.V. Artificial echo state network for predicting extraordinary situations in behavior of a rolling stock carriage equipment. *Nechetkie sistemy i myagkie vychisleniya* [Fuzzy Systems and Soft Computing]. 2013, vol. 8, no. 1, pp. 45–57 (in Russ.).
9. Ivanova E.I., Sorokin S.V. *Optimizator neyrosetevykh modeley prognozirovaniya neispravnostey oborudovaniya zheleznodorozhnykh vagonov* [Neural Network Models Optimizer to Predict Failure of Rolling Equipment on Railways]. PC program State Registr. Certif. no. 2014615050 (in Russ.).
10. Sorokin S.V., Sorokina I.V., Yazenin A.V. Equipment fault prediction using c neural networks and fuzzy inference. *Tekhnika zheleznykh dorog* [Railway Equipment]. 2014, no. 2 (26), pp. 64–74 (in Russ.).
11. Ivanova E.I. Methods for predicting wagon equipment malfunctions based on multi-layer neural network with genetic algorithm and neural echo-network: a comparative analysis. *Vestn. TvGU. Ser. Prikladnaya matematika* [Herald of Tver State Univ. Series: Applied Mathematics]. 2014, no. 2, pp. 67–79 (in Russ.).
12. *Plotting and Interpreting an ROC Curve*. Available at: <http://gim.unmc.edu/dxtests/roc2.htm> (accessed June 20, 2015).
13. *The Area Under a ROC Curve*. Available at: <http://gim.unmc.edu/dxtests/roc3.htm> (accessed June 20, 2015).

УДК 658:005.05: 519.859

Дата подачи статьи: 27.07.15

DOI: 10.15827/0236-235X.112.237-243

АЛГОРИТМЫ И МОДЕЛИ АСУ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

(Работа выполнена при финансовой поддержке РФФИ, проекты №№ 14-01-00324, 15-37-50431)

*Г.Б. Бурдо, д.т.н., зав. кафедрой, gbtms@yandex.ru
(Тверской государственный технический университет,
наб. Аф. Никитина, 22, г. Тверь, 170026, Россия);*

*А.В. Федотова, к.т.н., доцент, afedotova.bmstu@gmail.com
(Московский государственный технический университет им. Н.Э. Баумана,
ул. 2-я Бауманская, 5, г. Москва, 105005, Россия)*

Основные проблемы, возникающие при использовании АСУ технологическими процессами (ТП) в многономенклатурных производствах, к которым зачастую относятся производственные системы предприятий, выпускающих наукоемкую продукцию, обусловлены высокой динамичностью ситуации в фирмах. Это вызвано постоянным появлением новых контрактов в течение календарного года, что приводит к корректировке производственных планов предприятий, приближенностью технического нормирования из-за дефицита времени на технологическую подготовку, приводящую к неточности оперативного планирования.

На взгляд авторов, предлагаемая АСУТП (на примере технического обслуживания авиатехники) лишена указанных недостатков.

Это достигается за счет следующих факторов. Иерархическая структура АСУТП отражает иерархию производственных планов предприятия и представлена подсистемами объемного, календарного, оперативного планирования и диспетчирования. АСУТП как система в целом информационно связана с системой управления организации и является ее подсистемой.

Для обеспечения диспетчирования информационное взаимодействие осуществляется и с производственными подразделениями. Все подсистемы АСУТП информационно связаны, что позволяет оперативно выполнять корректировку всех видов планов. Планирование происходит путем использования итерационных процедур, обеспечивающих «тонкое» наполнение производственных планов, при этом за счет обратных связей существует возможность корректировки производственных планов высших уровней, что обеспечивает их выполнимость. При производственном планировании предусматриваются локальные резервы мощностей и фондов времени работы, предлагаются мероприятия по их использованию.

Работа подсистемы оперативного планирования, заканчивающаяся составлением расписания работы производственных подразделений, основана на аналитико-приоритетном методе. Показана уточненная постановка задачи составления расписания работы производственного подразделения, особенности связаны с выполнением работ бригадами рабочих и применением приоритетных схем обслуживания.

Выявлены и показаны способы формирования приоритетных схем обслуживания объектов, обеспечивающие вариативность расчетов и возможность выбора оптимального решения в соответствии с установленным критерием. При доработке АСУТП может быть реализована в качестве системы поддержки принятия решений на различных предприятиях многономенклатурного машиностроения, занимающихся обслуживанием, ремонтом и сборкой наукоемких изделий.

Ключевые слова: автоматизированные системы управления технологическими процессами, планирование, диспетчирование, наукоемкая продукция, искусственный интеллект, поддержка решений.

Известно, что применение автоматизированных систем управления технологическими процессами (АСУТП) в многономенклатурном машиностроении затрудняется необходимостью корректировки планов всех уровней вследствие динамичной производственной ситуации и постоянным появлением новых контрактов. Ситуация усугубляется для ТП с большой долей ручного труда, что приводит еще и к неопределенности времени выполнения неповторяющихся технологических операций. К таким технологическим процессам, в частности, относятся сборка изделий, техническое обслуживание изделий и их ремонт, изготовление новых образцов наукоемкой продукции.

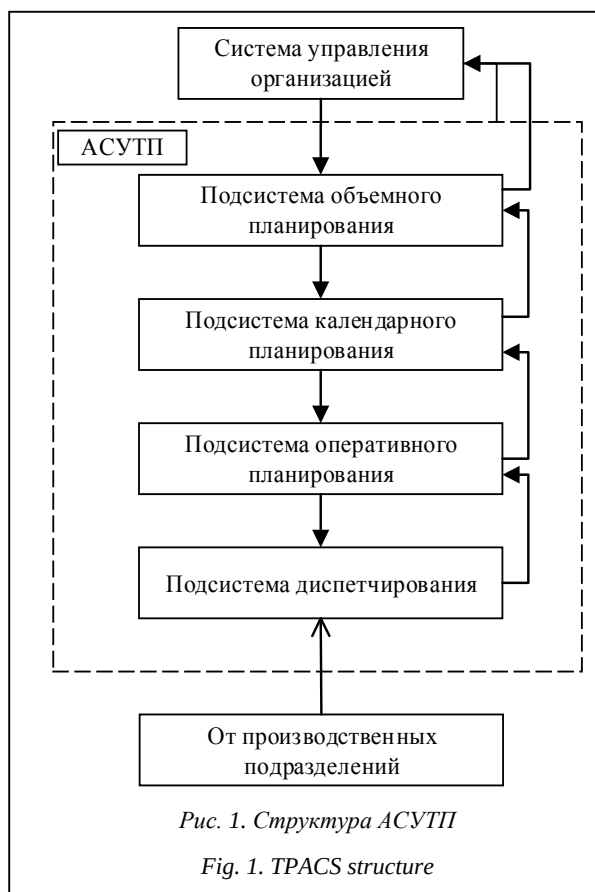
В статье рассмотрена АСУТП технического обслуживания авиационной техники, позволяющая минимизировать воздействие перечисленных негативных факторов.

В работах [1, 2] были представлены общие принципы создания АСУТП. Структура автоматизированной системы управления процессами обслуживания авиационной техники, соответствующая им, показана на рисунке 1. Иерархия подсистем АСУТП в силу системного принципа преемственности соответствует структуре производственных планов авиаремонтных предприятий, обратные связи между подсистемами обеспечивают возможность корректировки планов всех уровней [1], что обеспечивает их выполнимость.

Подсистема объемного планирования

Алгоритм работы подсистемы объемного планирования следующий.

На первом шаге оцениваются годовые мощности производственных подразделений (ПП), выпол-



няющих *техническое обслуживание* (ТО) авиатехники.

Должно быть:

$$\sum_{i=1}^n (S_i \cdot T_i) \leq 0,8 \cdot S_{\text{ПП}} \cdot F_r \cdot N, \quad (1)$$

где S_i – площадь, занимаемая i -м обслуживаемым объектом; n – их количество; T_i – трудоемкость обслуживания i -го объекта, час; $S_{\text{ПП}}$ – площадь ПП; F_r – номинальный годовой фонд времени работы ПП при односменной работе; N – число смен; 0,8 – коэффициент, учитывающий различие между трудоемкостью ТО и его циклом и необходимость наличия резервов.

$$T_i = \left(\sum_{j=1}^m t_{\text{шт.к}_j} \right)_i - \sum_{j=1}^m t_{\text{шт.к}_j}^c, \quad (2)$$

где $t_{\text{шт.к}_j}$ – штучно-калькуляционное время выполнения j -й операции над i -м объектом; $\sum_{j=1}^m t_{\text{шт.к}_j}^c$ – суммарное время совмещенных (параллельно выполняемых во времени) операций ТО над i -м объектом; m – число операций.

Если неравенство (1) выполняется, осуществляется переход к шагу 4 планирования, иначе переход к шагу 2.

На втором шаге итерационно выполняются корректировка объемного планирования и его согласование с СУО. Возможны следующие варианты.

Первый – удаление из плана часто обслуживаемых изделий до выполнения формулы (1) по следующим приоритетным правилам.

Правило 1. Из плана следует исключать наименьшее число изделий, обеспечивающих выполнение условия (1).

Правило 2. При возможных альтернативных вариантах (при различии между произведением $S_i \cdot T_i$ для исключаемых из плана объектов до 10 %) из плана ТО исключают объект, имеющий большее значение S_i .

Правило 3. Если различие между S_i для исключаемых объектов до 10 %, то из плана ТО исключают объект, имеющий меньшую договорную цену обслуживания.

Правило 4. Из плана не исключаются объекты, ТО которых безоговорочно должно быть выполнено (приоритет организации).

Выявленные правила исключения отражают смысл работы организации и ПП: получение максимального дохода и создание предпосылок эффективной работы.

Если удастся полностью откорректировать объемное планирование, осуществляется переход к шагу 4, иначе к шагу 3.

На третьем шаге делается попытка удовлетворить неравенство (1) путем увеличения фонда времени работы ПП за счет K дополнительных смен в течение года. Следует отметить, что данный шаг – прерогатива системы управления организацией, определяющей допустимое число дополнительных смен K_d . Должно выполняться

$$K_d \geq \frac{\sum_{i=1}^n (S_i \cdot T_i) - 0,8 \cdot S_{\text{ПП}} \cdot F_r}{8 \cdot S_{\text{ПП}}} = K, \quad (3)$$

где 8 – длительность смены в часах.

Если условие (3) удастся удовлетворить, осуществляется переход к шагу 4, иначе возврат на шаг 2.

На четвертом шаге проверяется выполнимость работ исходя из общих фондов времени работы ремонтников и по видам работ (при специализации видов работ за отдельными исполнителями). Должны выполняться следующие условия.

По общему фонду времени работ:

$$\sum_{i=1}^n T_i \leq P \cdot F_p \cdot 0,8 \cdot N, \quad (4)$$

где P – общая численность рабочих, осуществляющих техническое обслуживание; F_p – годовой фонд времени рабочего, час.

По фондам времени по отдельным видам работ для каждого из них m должно выполняться, $m = 1, B$:

$$\sum_{i=1}^n \left(\sum_{j=1}^1 t_{\text{шт.к}_m} \right)_i \leq P_m \cdot F_p \cdot 0,8, \quad (5)$$

где P_m – общее число рабочих, выполняющих работу вида m , $m = 1, B$.

Если рабочие, выполняющие ТО, специализируются на нескольких видах работ m ($m=1, p$), то должно выполняться условие

$$\sum_{i=1}^n (\sum_{m=1}^p t_{шт.к.m})_i \leq P_m \cdot F_p \cdot 0,8. \quad (6)$$

Если работа выполняется бригадами, то P_m – общее число бригад.

Если условия (4)–(6) выполняются, происходит переход к подсистеме календарного планирования, иначе на шаг 5.

На пятом шаге делается попытка выполнить неравенства (4)–(6) за счет введения дополнительных смен, при этом их число не может быть более K_d (см. описание шага 3).

Для случая (4):

$$K_d \geq (\sum_{i=1}^n T_i - P \cdot F_p \cdot 0,8) / 8 = K. \quad (7)$$

Для случая (5):

$$K_d \geq \sum_{i=1}^n (\sum_{m=1}^p t_{шт.к.m})_i - P_m \cdot F_p \cdot 0,8 / 8 = K. \quad (8)$$

Для случая (6):

$$K_d \geq \sum_{i=1}^n (\sum_{m=1}^p t_{шт.к.m})_i - P_m \cdot F_p \cdot 0,8 / 8 = K. \quad (9)$$

Для дальнейших расчетов принимается максимальное значение K , а если выполнялся шаг 3, то наибольшее значение (с учетом и полученного по формуле (3)).

Возможно увеличение суммарных фондов времени и за счет увеличения численности работающих P и P_m . Однако этот шаг в условиях рассматриваемых производственных систем не всегда реализуем из-за дефицита квалифицированных рабочих данного профиля и трудности прогнозирования последующих заказов.

Если условие выполняется, то переходят к подсистеме календарного планирования, иначе к шагу 6.

На шестом шаге производится исключение работ из годового объемного плана по следующим приоритетным правилам.

Правило 1. Из плана не исключаются изделия, ТО которых должно быть безоговорочно выполнено.

Правило 2. Из плана следует исключать наименьшее число изделий.

Правило 3. Исключением из плана изделий сначала обеспечивают выполнение неравенства (9), затем (8) и в последнюю очередь (7).

Правило 4. При возможных альтернативах (при различии в числе K не более 10 %) исключают из плана объект, имеющий меньшую договорную цену обслуживания.

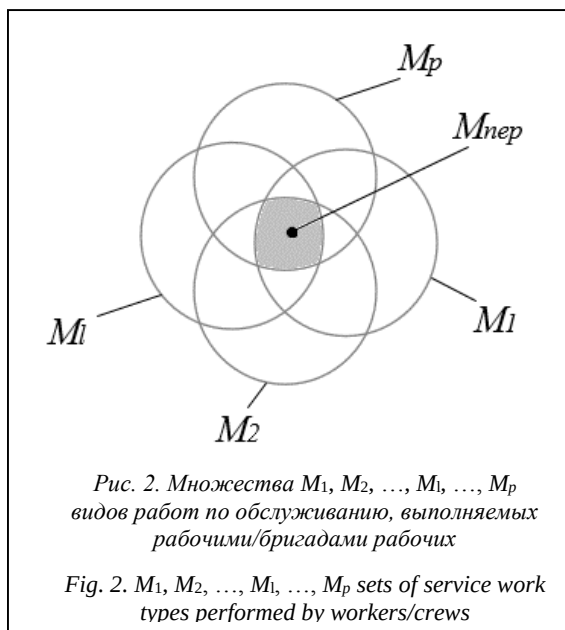
Возврат к шагу 4.

Указанные правила обеспечивают достаточно «точное» формирование объемного планирования и получение максимального дохода организации.

В целом особенностями работы рассматриваемой подсистемы являются достаточно высокая формальность правил выработки советующих управляющих воздействий, ясность процедур синтеза решений и проверка возможностей производственных подразделений не только по мощностям, но и по фондам времени работы рабочих.

Хотелось бы отметить и возможность использования в практике ТО такой возможности, как освоение работниками смежных видов работ, что создает высокую гибкость производственных систем. В пределе, когда каждый рабочий (бригада) может выполнять любую работу, необходима проверка на шаге 4 лишь по формуле (4), а на шаге 5 – по (7). Также будут упрощены и задачи, решаемые при календарном и оперативном планировании.

Поэтому целевой функцией при подборе и обучении рабочих производственных подразделений может являться мощность пересечения (рис. 2) множеств видов работ, выполняемых рабочими (бригадами): $M_{пер} \rightarrow \max$.



Подсистема календарного планирования

Работа подсистемы оперативного планирования также включает выполнение ряда шагов.

На первом шаге осуществляется предварительный набор заказов для первого календарного плана (обычно квартального). Формирование выполняется исходя из более ранних сроков окончания работ по обслуживанию объектов и приоритетности заказов. Набор заказов заканчивается при невозможности удовлетворить неравенство (1) при присоединении еще одного любого заказа. В формуле

вместо значения F_T подставляется $F_{кв}$ (квартальный фонд времени работы).

На *втором шаге* рассчитывается длительность технологических циклов выполнения работ по каждому i -му заказу:

$$\Pi_i = T_i + \Pi_i \cdot T_{п.ср} + t_{он}, \quad (10)$$

где Π_i – ожидаемое количество перерывов (ожиданий) при обслуживании i -го объекта; $T_{п.ср}$ – среднее время перерыва, Π_i и $T_{п.ср}$ определяются из опыта выполнения обслуживания; $t_{он}$ – среднее время ожидания начала обслуживания, тратится на высвобождение рабочего места, установку нового объекта и подготовку к обслуживанию.

На *третьем шаге* выполняется размещение объектов по календарным срокам выполнения работ. При этом объекты делятся на две группы: объекты с заданными предельными сроками и объекты с окончанием работ в пределах всего календарного периода (рис. 3). Обозначения на рисунке: $S_1, S_2, S_{ин}$ – площади, занимаемые обслуживаемыми объектами; $S_{пп}$ – суммарная площадь производственных подразделений; Π_1, Π_2, Π_m – технологические циклы обслуживания объектов; $t_{ок1}, t_{ок2}$ – предельные сроки обслуживания 1-го и 2-го объектов. Как можно увидеть, для объектов 1-й группы предусмотрен резерв времени.

К примеру, i -й объект может быть обслужен в пределах всего календарного срока, его фактический цикл Π_i^{ϕ} разбивается на две части, при этом он больше Π_i (см. формулу (10)) на величину $t_{ож}$. Такой подход обеспечивает более полное использование мощностей предприятия.

Для построения пространственно-временной структуры загрузки производственных подразделений используются алгоритмы, аналогичные алго-

ритмам листового раскроя материалов, на которые накладываются ограничения (в данном случае S_i и Π_i ассоциируются с длиной и шириной заготовки, $S_{пп}$ и F_k – ширина листа) следующего вида.

1. Если m -й объект имеет оговоренные сроки завершения обслуживания $t_{ок,m}$, то оно должно быть завершено с опережением на l дней (обычно $l = 2-3$ дня), то есть $t_{ок}^m - t_{ок,m} = 2-3$ дня.

2. Площадь, не закрытая объектами, не должна превышать 10 % от общей.

3. Если условие 2 не удовлетворяется и остаются необслуженные объекты, то для объектов 2-й группы выполняется разделение технологического цикла на этапы по следующим правилам.

Правило 1. Первыми разделение выполняют для объектов с наиболее длительным циклом.

Правило 2. Первым заполняется свободное рабочее место подходящей площади, расположенное наиболее близко к началу календарного периода (то есть слева на рис. 3).

Правило 3. Разделение цикла выполняют с обеспечением целостности операций (см. формулу (2)) до приближения выделенной части цикла к длительности незанятости рабочего места.

Далее алгоритм распределения запускают вновь.

3. Если условие 2 удовлетворяется после разбиения циклов и есть необслуженные объекты, то переход к шагу 4.

4. Если условие 2 не удовлетворяется и нет необслуживаемых объектов, то переход к шагу 5.

5. Если условие 2 удовлетворяется и нет необслуженных объектов, то переход к шагу 6.

На *четвертом шаге* для обеспечения выполнения планов анализируется возможность введения дополнительных смен, то есть увеличение величины F_k (см. формулу (3), K_d берется в расчете на квартал).

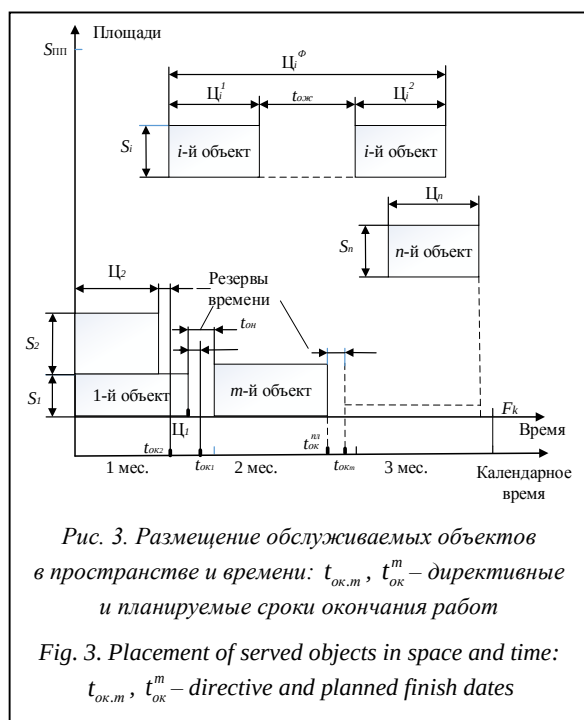
При отсутствии возможности заказы переносятся на следующий квартал. Переход к шагу 6.

На *пятом шаге* анализируется возможность обслуживания объектов из следующего календарного периода в данном. Эти объекты относят ко 2-й группе в данном календарном периоде. В этом случае из следующего календарного периода берутся объекты с более ранними сроками выполнения и возвращаются на шаг 2.

При корректировке пространственно-временной структуры на шаге 3 ранее запланированные объекты не трогаются, заполняются лишь свободные площади.

На *шестом шаге* выполняется проверка по фондам времени рабочих, выполняющих техническое обслуживание по каждому месяцу календарного плана.

Проверка выполняется по формулам, аналогичным (5) и (6), вместо расчетного годового времени подставляется фонд времени F_M (фонд времени за месяц периода календарного планирования). Если



планы выполнимы для всех месяцев, то переходят к подсистеме оперативного планирования, в противном случае выполняют попытки создать дополнительное число рабочих смен (см. формулы (8) и (9), Кд берется в расчете на месяц). Если данное мероприятие удастся реализовать, то переходят к подсистеме оперативного планирования, иначе – к шагу 7.

На *седьмом шаге* выполняется выделение из месячного календарного плана обслуживания объектов до выполнения условия, оговоренного на предыдущем шаге, по следующим приоритетным правилам, и перенос их обслуживания на следующий месяц.

Правило 1. Из плана не выделяются объекты 1-й группы (с жестко оговоренным сроком окончания обслуживания).

Правило 2. Выделение начинают с объекта, имеющего минимальную трудоемкость обслуживания.

Правило 3. Если невозможно удовлетворить условиям (см. шаг 6) выделением одного объекта, то переходят к выделению второго и т.д. При удовлетворении условий происходит возврат на шаг 3.

Подсистема оперативного планирования

Синтезированные месячные календарные планы работ служат основой для оперативного планирования и управления работой технологического подразделения, выполняющего ТО. С этой целью покажем формальную постановку задачи составления *календарного плана-графика* (КПГ) (расписания) работы подразделения, в котором выполняется ТО. Постановка сделана на основе работ [2–5]. КПГ разрабатываются на интервалы *оперативного планирования* (ОП), равные 5 календарным дням. Таким образом, месячный календарный план работ должен быть представлен как совокупность КПГ.

Пусть задано подразделение, выполняющее ТО с установившимся режимом работы. В подразделении должно быть выполнено ТО некоторого множества объектов $O=\{O_i\}$, $i=\overline{1,d}$; d – количество объектов.

По каждому из объектов должна быть выполнена процедура $ПР_i$. Каждая процедура $ПР_i$ состоит из множества операций $\{ОП_{ij}\}$, $j=\overline{1,m_i}$, где m_i – количество операций обслуживания i -го объекта.

Каждой операции $ОП_{ij}$ присваивается индекс k из множества $\{1, 2, \dots, l\}$, характеризующий вид выполняемой работы в операции (вид операции), то есть $ПР_i = \{ОП_{ijk}\}$.

Каждая операция характеризуется трудоемкостью ее выполнения, равной штучно-калькуляционному времени ее выполнения $T_{ij} = t_{шт.к.ij}$.

Обслуживание выполняется множеством R бригад рабочих, $R=\{r_m\}$; $m=\overline{1,c}$, где c – количество бригад.

Каждая бригада r_m выполняет p операций $ОП_m^r$ определенного вида

$$ОП_m^r = \{ОП_{mn}^r\}; m=\overline{1,c}; n=\overline{1,l^r}, \text{ где } l^r - \text{множе-}$$

ство видов операций, выполняемых бригадой.

Для обеспечения выполнимости операций $\{ОП_{ijk}\}$ должно быть выдержано условие, означающее, что все виды операций могут быть выполнены исполнителями:

$$[G = U(1, 2, \dots, l)] \subset G^r = \cup(1, 2, \dots, l^r).$$

Порядок следования операций $\{ОП_{ijk}\}$ в процедуре $ПР_i$ задается следующим образом.

Задаются булевы переменные α и β . Для $\{ОП_{ijk}\}$

$$\alpha_j = \begin{cases} 0, & \text{если операция независима} \\ & \text{и может выполняться в любом месте} \\ & \text{процедуры, в том числе параллельно} \\ & \text{с другими;} \\ 1, & \text{если порядок следования оговаривается.} \end{cases}$$

Для двух операций j' и j'' с $\alpha_{j'}=1$ и $\alpha_{j''}=1$ значения

$$\beta_{jj''} = \begin{cases} 0, & \text{если операции могут выполняться} \\ & \text{параллельно,} \\ 1, & \text{если операция } j' \text{ предшествует} \\ & \text{операции } j'' \text{ (операции несовместимы} \\ & \text{во времени).} \end{cases}$$

Ограничения на формируемый КПГ следующие.

1. Непрерывность выполнения работ над объектом. Для этой цели время начала t_{ij}^o и окончания t_{ij}^k работы на $ОП_{ijk}$ связывается соотношением $t_{ij}^k = t_{ij}^o + T_{ij}$.

2. Невозможность параллельного выполнения на одном объекте несовместимых работ, то есть для них условие $t_{ij}^o \geq t_{ij'}^k$, операция j' предшествует j'' (операции несовместимы). Это новое ограничение по сравнению с известным.

3. Невозможность одновременного проведения числа операций, большего числа бригад, то есть для любого множества операций $\{ОП_{ij}\}=\{ОП_{ij}, \dots, ОП_{ij'}\}$ условие $t_{ij}^o \leq t_{ij'}^o \leq t_{ij}^k$ не может выполняться более c раз (c – число бригад).

4. Невозможность одновременного проведения операций одинаковых видов, большего числа специализированных для этих операций бригад. Для учета этого ограничения введем переменную

$$X_{ijl} = \begin{cases} 1, & \text{если вид технологической операции } ОП_{ijk} \\ & \text{совпадает с видом операций, выполняемых} \\ & \text{рабочими } ОП_{mn}, \text{ то есть } k=n, \\ 0 & \text{в противном случае.} \end{cases}$$

В любой момент времени работы подразделения $t_{ij}^o \leq t \leq t_{ij}^k \sum X_{ijl} \leq B$, где B – число бригад, которые могут выполнять работу группы n .

5. Учитывая, что, как правило, реальная производственная ситуация, складывающаяся в ПП, динамична и не всегда формальна, введем ряд огра-

ничений – приоритетов, отражающих текущее состояние подразделений и директивы управленческого персонала. В общем виде его можно сформулировать так: на объекте с более высоким приоритетом в первую очередь выполняется операция освободившейся бригадой, то есть $t_{ijk}^o < t_{i'jk}^o$, если приоритет объекта i выше i' . Это новое ограничение по сравнению с известными постановками.

Применение приоритетов во многом упрощает алгоритмы построения КПП.

Требуется построить $KPP = \{t_{ijk}^o\}$, отвечающий ограничениям, перечисленным выше. В качестве целевой функции выступает общий минимальный цикл ТО всех объектов, $T_o: T_o = \max_{ijk} \{t_{ijk}^o\} \rightarrow \min$.

Генерация КПП выполняется на основе известного аналитико-приоритетного метода с использованием пошаговых процедур (имитации) выполнения операций ТО над объектами [3–9].

В качестве правил, определяющих приоритеты обслуживаемых объектов, могут быть следующие [10].

1. Установление приоритетов объектов руководителем подразделения. В этом случае каждому объекту присваивается ранг, скажем, цифрами от 1 до d , где d – общее число объектов. Учитывая, что на объекте одной и той же бригадой может быть выполнено несколько работ, для получения вариантов КПП следует в одном варианте предусмотреть дополнительный приоритет выполнения операции максимальной трудоемкости, а во втором минимальной.

2. Установление приоритетов нескольким объектам, остальные имеют приоритеты ниже первых. Для них приоритеты могут быть установлены следующим образом, определяющим два варианта возможных КПП:

- первым освободившейся бригадой обслуживается объект с минимальной остаточной суммарной трудоемкостью, то есть объект с максимальным приоритетом; этот приоритет исходит из посыла, что лучше быстрее закончить ТО тех объектов, которые имеют меньший объем остаточных работ;

- выровнять объекты проведенного ТО по всем объектам: первым освободившейся бригадой выполняется обслуживание объекта с максимальной остаточной суммарной трудоемкостью операций.

Учитывая, что по наиболее приоритетному объекту может быть выполнено несколько операций, смыслу приоритетов будет отвечать выбор операции максимальной трудоемкости.

3. Если приоритеты обслуживания не назначаются руководством подразделения, по вышеприведенным правилам приоритеты назначаются для всех объектов.

Подсистема диспетчирования

Диспетчирование работ осуществляется путем контроля за ходом ТО объектов сравнением планового и фактического времени окончания операций, то есть при этом анализируются следующие данные:

- отставание (опережение) выполнения КПП по подразделению, час;
- изменение отставания (опережения) КПП в данном интервале ОП по сравнению с предыдущим (то есть тенденция изменения), час/смена;
- фактические и действительные фонды времени выполняющих обслуживание рабочих (бригад);
- расчетные нормы и фактическое время выполнения операций обслуживания, час;
- расчетный и фактический коэффициенты параллельного выполнения операций, %.

Продукционными моделями (в силу громозкости не приводятся) на основе указанных данных вырабатываются варианты управленческих решений по приведению фактических КПП к расчетным, в том числе и пересчет ранее сформированных КПП и календарных планов.

В заключение отметим, что указанная автоматизированная система разрабатывалась как советующая, помогающая оперативно принимать решения на всех уровнях планирования и имеющая элементы искусственного интеллекта. Ее отличительными особенностями от известных систем класса «Scada» и ERP [11] являются, во-первых, реализация оперативной связи между планами всех уровней и возможность их комплексного пересчета при появлении новых заказов (заключение контрактов в течение года) с различными приоритетами, во-вторых – высокая степень формальности процедур и, как следствие, точность рекомендуемых решений, в-третьих – возможность обучения путем анализа расчетных норм и фактического времени выполнения операций и расчетного и фактического коэффициентов параллельного по времени выполнения операций.

При доработке АСУТП может быть реализована в качестве системы поддержки принятия решений на различных предприятиях многоименного машиностроения, занимающихся обслуживанием, ремонтом и сборкой наукоемких изделий.

Литература

1. Бурдо Г.Б. Принципы построения автоматизированной системы управления технологическими процессами в многоименных производствах // Вестн. Саратовского гос. техн. ун-та. 2010. № 3 (48). С. 113–118.
2. Бурдо Г.Б. Интеллектуальные процедуры планирования и управления в производственных системах геофизического приборостроения // Программные продукты и системы. 2011. № 3 (95). С. 107–110.
3. Прилуцкий М.Х., Власов В.С. Метод ветвей и границ с

эвристическими оценками для конвейерной задачи теории расписаний // Вестн. Нижегородского ун-та им. Н.И. Лобачевского. 2008. № 3. С. 143–157.

4. Смоляр Л.И. Модели оперативного планирования в дискретном производстве. М.: Наука, 1988. 320 с.

5. Прилуцкий М.Х. Многокритериальные многоиндексные задачи объемно-календарного планирования // Изв. АН. Теория и системы управления. 2007. № 1. С. 78–82.

6. Прилуцкий М.Х., Батищев Д.И., Гудман Э.Д., Норенков И.П. Метод декомпозиции для решения комбинаторных задач упорядочения и распределения ресурсов // Информационные технологии. 1997. № 1. С. 29–33.

7. Султан-заде Н.М., Загидуллин Р.Р. Повышение произ-

водительности ГПС путем оптимизации расписаний. М.: СТИН, 1996. № 12. С. 9–13.

8. Танаев В.С., Шкурба В.В. Введение в теорию расписаний. М.: Наука, 1975. 256 с.

9. Блехерман М.Х. Оптимизация загрузки оборудования автоматизированных участков из станков с ЧПУ // Станки и инструмент. 1979. № 5. С. 3–5.

10. Бурдо Г.Б. Оперативное планирование на основе приоритетных схем в производственных системах геофизического приборостроения // Каротажник. 2011. № 6 (204). С. 92–100.

11. Балахонова И.В., Волчков С.А., Капитуров В.А. Интеграция процессов с помощью ERP-системы. СПб: Приоритет, 2006. 464 с.

DOI: 10.15827/0236-235X.112.237-243

Received 27.07.15

ALGORITHMS AND MODELS FOR MAINTENANCE PROCESS AUTOMATED CONTROL SYSTEMS

(The work has been done with financial support from RFBR, projects no. 14-01-00324, 15-37-50431)

Burdo G.B., Dr.Sc. (Engineering), Head of Chair, gbtms@yandex.ru

(Tver State Technical University, Nikitin Quay 22, Tver, 170026, Russian Federation);

Fedotova A.V., Ph.D. (Engineering), Associate Professor, aluon@mail.ru

(Bauman Moscow State Technical University, 2nd Baumanskaya St. 5, Moscow, 105005, Russian Federation)

Abstract. Production systems of enterprises, which produce high technology products, often relate to multiproduct production. When using the technological processes automated control systems (TPACS) in multiproduct production, the main problems are caused by the high situation dynamism in the companies. This is due to the regular emergence of new contracts during a calendar year, resulting in production plans adjustment; the technical regulation approximation because of lack of time for technological preparation leading to operational planning inaccuracy.

According to the authors, the proposed process automated control system (for example, aircraft technical maintenance) is devoid of these shortcomings. This is due to the following factors. The hierarchical structure of TPACS reflects the hierarchy of the enterprise production plans and is presented by subsystems of volume, calendar, operational planning and scheduling. TPACS as a system in whole is connected to an organization management system through information and is its subsystem.

To ensure scheduling, there is information interaction with production departments. All the subsystems of TPACS are connected, so it is possible to correct all types of plans quickly. Planning is accomplished by using iterative procedures, which provide “thin” fulfillment of production plans. Therefore, higher level production plans adjusting is possible due to feedbacks, and this ensures their feasibility. When production planning, local capacity reserves and operating time funds are planned, the ways of using their reserves are suggested.

The operational planning subsystem results in scheduling of production units. It is based on the relative priority method. The paper shows the refined statement of the production unit schedule problem. The particular features are associated with the performance of work by work groups and with the use of priority service schemes.

The paper identifies the methods of forming the priority object service schemes that provide a variety of calculations and a choice of optimum in accordance with the established criterion. When refining, TPACS can be implemented as a decision support system in various diversified engineering enterprises, which provide service, repair and high technology products assembly.

Keywords: technological processes automated control systems, scheduling, dispatching, high technology products, artificial intelligence, decision support.

References

1. Burdo G.B. Constructing principles of technological processes automated control systems in multiproduct manufacturing. *Vestnik Saratovskogo gos. tekhnich. univ.* [Vestnik Saratov State Technical Univ.]. 2010, no. 3 (48), pp. 113–118 (in Russ.).
2. Burdo G.B. Intelligence plan and management of production systems in geophysical manufacture. *Programmnye produkty i sistemy* [Software & Systems]. 2011, no. 3 (95), pp. 107–110 (in Russ.).
3. Prilutsky M.Kh., Vlasov V.S. The method of paths and boundaries with heuristic evaluation for a conveyor problem of scheduling theory. *Vestnik Nizhegorodskogo univ. im. N.I. Lobachevskogo* [Vestnik of Lobachevsky University of Nizhni Novgorod]. 2008, no. 3, pp. 147–153 (in Russ.).
4. Smolyar L.I. *Modeli operativnogo planirovaniya v diskretnom proizvodstve* [Operating Planning Models In Discrete Production]. Moscow, Nauka Publ., 1988, 320 p.
5. Prilutsky M.Kh. Multicriterial Multi-Index Resource Scheduling Problems. *Izvestiya AN. Teoriya i sistemy upravleniya* [Journ. of Computer and Systems Sciences International]. 2007, no. 1, pp. 78–82 (in Russ.).
6. Prilutsky M.Kh., Batischev D.I., Gudman E.D., Norenkov I.P. Decomposition method to solve recourses arrangement and distribution combinatory problems. *Informatsionnye tekhnologii* [Information Technologies]. 1997, no. 1, pp. 29–33 (in Russ.).
7. Sultan-zade N.M., Zagidullin R.R. *Povyshenie proizvoditelnosti GPS putem optimizatsii raspisany* [Improving GPS Performance by Optimizing Schedules]. Moscow, STIN Publ., 1996, no. 12, pp. 9–13 (in Russ.).
8. Tanaev V.S., Shkurba V.V. *Vvedenie v teoriyu raspisany* [Introduction to the Theory of Scheduling]. Moscow, Nauka Publ., 1975, 256 p.
9. Blekherman M.Kh. Production load optimization of automated cells with CNC machines. *Stanki i instrument* [Russian Engineering Research]. 1979, no. 5, pp. 3–5 (in Russ.).
10. Burdo G.B. Operating planning based on priority schemes in geophysical industry production systems. *Karotazhnik*. 2011, no. 6 (204), pp. 92–100 (in Russ.).
11. Balakhonova I.V., Volchikov S.A., Kapituov V.A. *Integratsiya protsessov s pomoshchyu ERP-sistemy* [Process integration using ERP-system]. St.-Petersburg, Prioritet Publ., 2006, 464 p.

УДК 004.416.6

Дата подачи статьи: 23.06.15

DOI: 10.15827/0236-235X.112.244-248

ПРОГРАММНЫЙ КОМПЛЕКС ПРОЕКТИРОВАНИЯ ОБОГРЕВА ТЕХНОЛОГИЧЕСКИХ ТРУБОПРОВОДОВ ТЕПЛОВЫМИ СПУТНИКАМИ ДЛЯ СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

Т.А. Кохов, аспирант, tkohov@ggo.ru

(АО «Гипрогазоочистка», ул. Первомайская, 126, г. Москва, 105203, Россия);

А.Б. Корельштейн, к.ф.-м.н., зам. директора, korelstein@truboprovod.ru

(НТП «Трубопровод», ул. Плеханова, 7, г. Москва, 111141, Россия);

Т.Н. Гартман, д.т.н., профессор, зав. кафедрой, gartman@muctr.ru

(Российский химико-технологический университет им. Д.И. Менделеева,

Миусская пл., 9, г. Москва, 125047, Россия)

В промышленных установках используются следующие виды обогрева технологических трубопроводов: обогрев трубопроводов обогревающими спутниками (теплоспутниками), обогрев трубопроводов в паровой рубашке, электрообогрев трубопроводов.

В зависимости от технических требований или от климатических условий в проекте может преобладать один из этих видов обогрева.

Порядка 10–15 % работы проектировщика (в сфере монтажного проектирования) приходится на проектирование обогрева технологических трубопроводов.

В некоторых случаях практически 40 % трубопроводов в проекте должны обогреваться теплоспутниками, а иногда этот показатель может достигать порядка 50 %. При этом примерно 80 % трудозатрат на проектирование обогрева приходится именно на проработку трасс обогревающих спутников в 3D-модели.

С целью ускорения проектирования обогрева технологических трубопроводов тепловыми спутниками был разработан программный комплекс проектирования обогрева технологических трубопроводов для систем автоматизированного проектирования, реализованный в среде 3D-проектирования AVEVA PDMS.

Отличительная особенность созданного программного комплекса проектирования обогрева технологических трубопроводов в сравнении с имеющимися системами в том, что она интегрирована в систему автоматизированного проектирования AVEVA PDMS. Это позволяет выполнять моделирование обогревающих спутников технологических трубопроводов в 3D-модели и выпускать необходимые конечные документы – изометрические чертежи трубопроводов.

Функциональные возможности программного комплекса проверены при разработке водяного обогрева технологических трубопроводов тепловыми спутниками установки производства серы.

Ключевые слова: тепловой спутник (теплоспутник), запитка, распитка, перекидка, распределительная гребенка.

В статье описывается программный комплекс проектирования обогрева технологических трубопроводов (ОТТ) тепловыми спутниками для систем автоматизированного проектирования (САПР), реализованный в среде 3D-проектирования AVEVA PDMS.

В промышленных установках используются следующие виды ОТТ:

- обогревающими спутниками (теплоспутниками);
- в паровой рубашке;
- электрообогрев (см. ВСН 2-82. Указания ...).

В зависимости от технических требований или от климатических условий в проекте может преобладать один из этих видов обогрева.

Порядка 10–15 % работы проектировщика (в сфере монтажного проектирования) приходится на проектирование ОТТ.

В настоящее время авторам известны следующие системы для проектирования ОТТ.

- Система проектирования VisiTrace 3D. Thermon и Jacobs-Nederlands презентовали данную систему на Европейской конференции IEEE/PCIS в 2007 году [1]. Система поддерживает интеграцию с современными системами трехмерного проекти-

рования трубопроводов, однако предназначена для проектирования электрообогрева.

- CompuTrace® ST (Thermon) – программа проектирования ОТТ с использованием теплоспутников [2]. Информация об ее интеграции с современными системами трехмерного проектирования трубопроводов в публичном доступе отсутствует.

- TraceCalc Pro – ПО для проектирования обогрева. TraceCalc Pro обеспечивает проектные расчеты (потери, нагрузки, температура), рекомендации для управления и контроля систем обогрева, получение отчетов [1]. О возможности интеграции с современными САПР авторам неизвестно.

- ChromaTrace 3.0 – ПО для проектирования обогрева. Позволяет настраивать различные сценарии процесса для поиска наиболее экономичного решения. Имеет визуальную структуру для труб, резервуаров и контрольных панелей, осуществляет обработку данных с помощью импорта и экспорта Microsoft Excel, все расчеты производит в соответствии с последней версией стандарта IEEE515, полученные отчеты мгновенно экспортирует в Microsoft Excel и Word или в формате PDF Adobe Acrobat, отображает назначения линии обогрева [3].

- TRACER LYNX – инструмент автоматизированного проектирования электрообогрева, использующий стандартное ПО в качестве платформы для взаимодействия непосредственно с большинством САПР [4].

Таким образом, все известные авторам существующие программные системы проектирования обогрева с помощью тепловых спутников не поддерживают (по имеющейся у авторов информации) интеграцию с современными системами трехмерного проектирования.

В некоторых случаях практически 40 % трубопроводов в проекте должны обогреваться тепло-спутниками, а иногда этот показатель может достигать порядка 50 %. При этом примерно 80 % трудозатрат на проектирование обогрева заключается именно в проработке трасс обогревающих спутников в 3D-модели.

Такие большие объемы работ требуют качественного и удобного инструмента для автоматизации выполнения проекта обогрева. Его отсутствие в процессе разработки проекта обогрева выявило следующие проблемы, которые сильно затрудняли процесс выполнения работ при проектировании трасс тепловых спутников в 3D-модели:

- сложности при выполнении проекта обогрева несколькими исполнителями из-за отсутствия возможности графического отображения трасс обогревающих спутников в модели;
- трудности контроля изменений в трассе теплоспутника из-за отсутствия связи между местами запиток и отпиток по всей длине трассы (несвязность контура обогрева) и связи между гребенками, перекидками и обогреваемыми трубами;
- большие трудозатраты на получение отчетов, содержащих всю необходимую информацию (тип обогрева, длина, число и диаметр спутников, данные об изоляции) о трассе теплоспутника.

С целью ускорения проектирования ОТТ тепловыми спутниками и устранения вышеуказанных проблем был разработан программный комплекс проектирования ОТТ для САПР AVEVA PDMS, общую структуру которого можно представить в виде двух модулей:

- модуля проектирования обогревающих спутников технологических трубопроводов непосредственно в рабочей модели (с применением модуля Design AVEVA PDMS);
- модуля автоматического выпуска необходимых конечных документов марки ОТТ: аксонометрических чертежей марки ОТТ (с применением модуля Draft AVEVA PDMS).

Процесс проектирования обогрева трубопроводов обогревающими спутниками требует решения целого ряда задач, связанных с взаимным расположением обогревающих и обогреваемых труб: оптимизация трассы обогрева путем уменьшения числа перекидок и увеличения количества обогреваемых труб в одном контуре обогрева, распределение си-

стемы подвода и отвода теплоносителя, комплексный учет систем обогрева разных трубопроводов [5–9].

Функциональные возможности программного комплекса проверены при разработке водяного ОТТ тепловыми спутниками установки производства серы (см. <http://www.swsys.ru/uploaded/image/2015-4-dop/8.jpg>).

Структура алгоритма системы проектирования ОТТ теплоспутниками представлена на рисунке 1.

Исходными данными для проектирования обогревающих спутников технологических трубопроводов в рабочей модели в соответствии с СТО ИПН/НТП 17-02-2010, ВСН 168-76/ММС СССР и другими нормативными документами являются диаметр обогреваемого трубопровода, число обогревающих спутников, диаметр обогревающего спутника, температура окружающего воздуха (для выбора типа обогрева), местонахождение трубопровода.

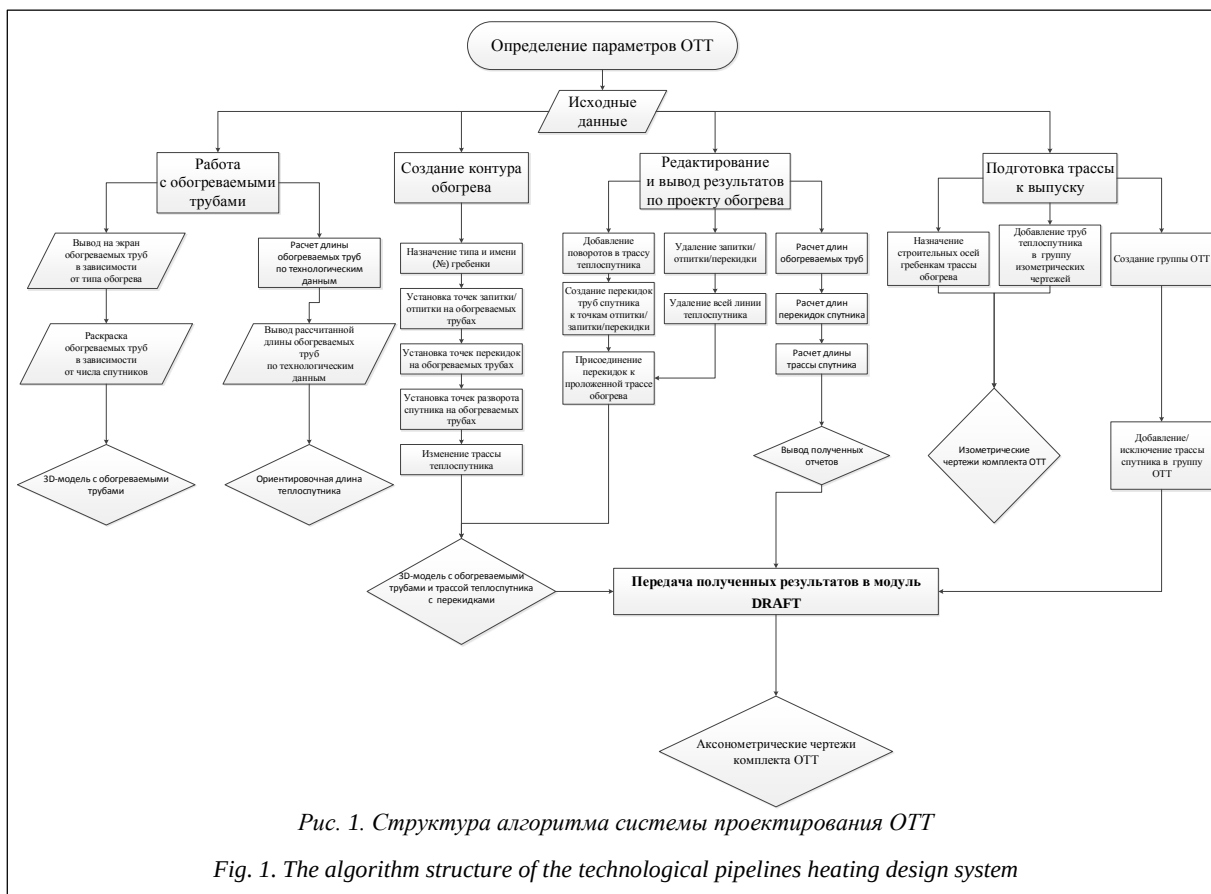
Реализация алгоритма системы проектирования ОТТ теплоспутниками осуществлялась на внутреннем языке САПР AVEVA PDMS – макроязыке программирования PML (Programmable Macro Language) [10].

Способ расстановки запиток/отпиток/перекидок основан на использовании режима Event-Driven Graphics (EDG): указателем мыши выбирается часть графического элемента в модели.

Отличительная особенность созданного программного комплекса проектирования ОТТ в сравнении с имеющимися системами в том, что он интегрирован в САПР AVEVA PDMS. Это позволяет выполнять моделирование обогревающих спутников технологических трубопроводов в 3D-модели (модуль Design) и выпускать необходимые конечные документы: изометрические чертежи трубопроводов (модуль IsoDraft) и чертежи марки ОТТ (модуль Draft).

Ключевые преимущества предлагаемого программного комплекса:

- сокращение трудозатрат на разработку проекта ОТТ в 1,2–1,5 раза;
- возможность выполнения проекта обогрева на завершающей стадии выполнения проектной документации, что позволяет учесть все изменения в проекте;
- отслеживание изменений при проектировании ОТТ и автоматическая генерация подробных отчетов по созданным трассам теплоспутников с учетом вносимых изменений;
- возможность выполнения ОТТ в рамках одного проекта несколькими исполнителями: каждый пользователь может работать со своей трассой теплоспутника, не пересекаясь с другими исполнителями, любой проектировщик может при необходимости продолжить проектирование трассы теплоспутника другого исполнителя, легко разобравшись в ней;



– возможность загрузки для работы всей системы труб (гребенки, перекидки, обогреваемые трубы), обогреваемых одной запиткой (одним штуцером на гребенке), что значительно облегчает работу проектировщика при продолжении выполнения проекта обогрева;

– возможность автоматического выпуска конечных документов отдельным комплектом ОТТ (аналогично комплекту TXM – чертежи трасс обогрева выполняются в аксонометрическом виде с применением модуля Draft CAIP AVEVA PDMS) (<http://www.swsys.ru/uploaded/image/2015-4-dop/10.jpg>); на одном листе отображается один контур обогрева, кроме случаев обогрева непосредственно от коллектора; на чертежах имеются следующие обозначения и информация: позиции обогревающих трубопроводов, позиции обогреваемых трубопроводов, обозначение гребенок, наименования оборудования, номера запиток/распиток, информация о прокладке теплоспутника в общей изоляции с обогреваемой трубой, диаметр и число спутников, таблица порядка обогрева трубопроводов, направление севера завода.

Функционально алгоритм системы проектирования ОТТ можно разделить на четыре основных блока.

Блок 1. Работа с обогреваемыми трубами. Блок позволяет получить необходимую информацию перед началом проектирования (интерфейс блока 1 представлен на рисунке 2):

– выбор обогреваемых труб в зависимости от типа обогрева;

– подсветка трубопроводов в зависимости от количества спутников;

– выполнение оценочных расчетов длин спутников и числа переходов.

Блок 2. Создание контура обогрева. Блок используется для создания контура обогрева (интерфейс блока 2 представлен на рисунке 2):

– назначаются тип и номер гребенки;

– строятся трасса первой перекидки от запиточной гребенки и трассы последней перекидки к распиточной гребенке;

– строятся все остальные перекидки;

– устанавливается место разворота теплоспутника при обогреве двумя или более спутниками;

– добавляются при необходимости новые участки обогрева к уже созданному контуру.

Блок 3. Редактирование/результаты (интерфейс блока 3 представлен на рисунке 2). Назначение блока:

– выбор способа графического отображения трассы теплоспутника;

– добавление поворотов в трассу теплоспутника;

– удаление всего контура или отдельно системы запитки/распитки;

– присоединение перекидок к местам запитки/распитки;

– получение необходимых отчетов.

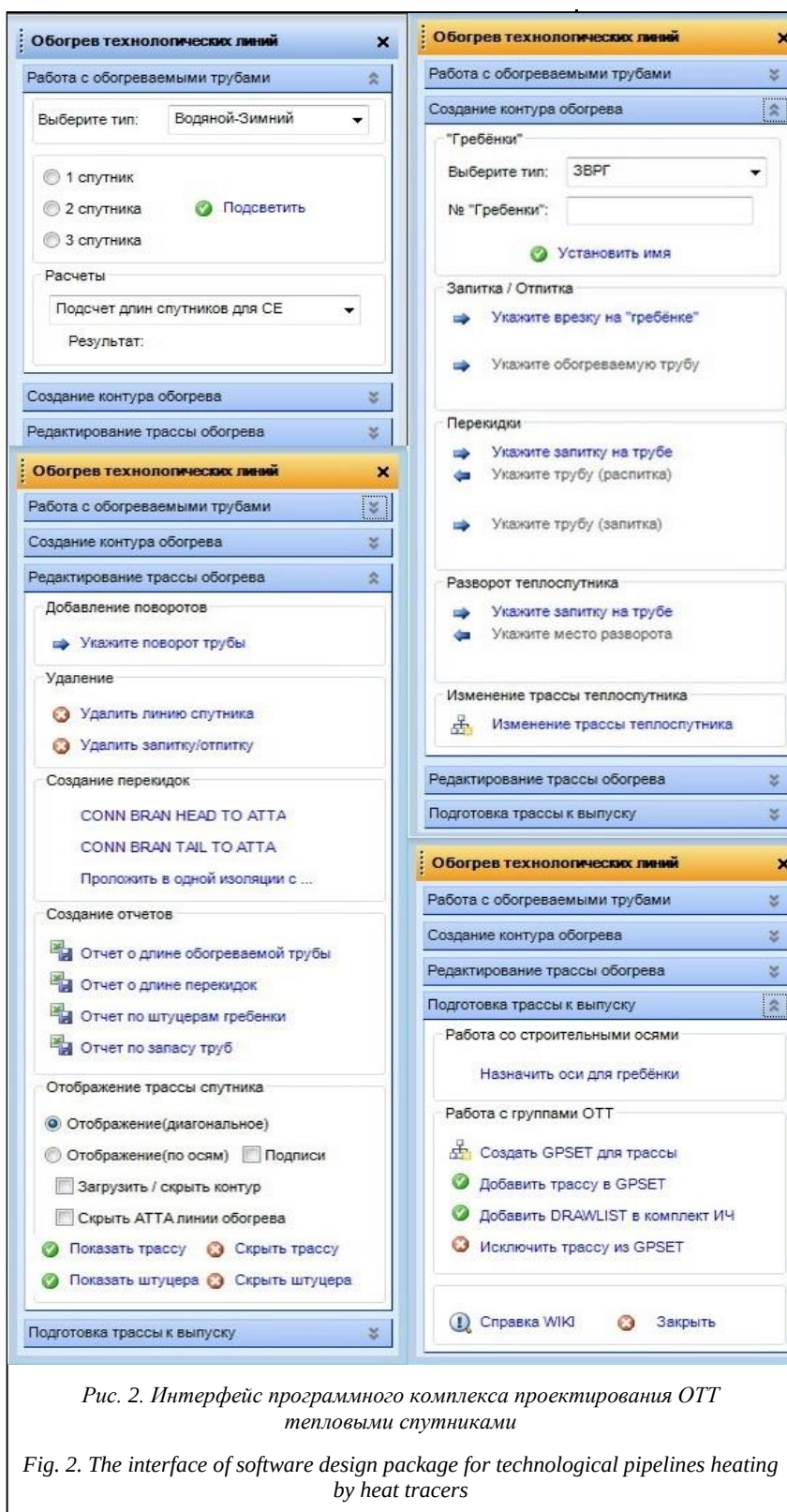


Рис. 2. Интерфейс программного комплекса проектирования ОТТ тепловыми спутниками

Fig. 2. The interface of software design package for technological pipelines heating by heat tracers

Блок 4. Подготовка трассы к выпуску. Блок для создания структуры проекта обогрева и подготовки трассы к выпуску (интерфейс блока 4 представлен на рисунке 2) используется

– для отображения привязок к строительным осям на изометрических чертежах гребенок;

– для структурирования и компоновки трубопроводов обогрева перед выпуском.

Разработанный программный комплекс проектирования ОТТ теплоспутниками для САПР AVEVA PDMS внедрен и успешно используется при разработке обогрева в проектах, выполняемых АО «Гипрогазоочистка».

Созданный программный комплекс проектирования ОТТ интегрирован в САПР AVEVA PDMS. Это позволяет выполнять моделирование обогреваемых спутников технологических трубопроводов и выпускать необходимые конечные документы.

Трудозатраты на разработку проекта ОТТ сократились в 1,2–1,5 раза.

Разработанный программный комплекс позволяет выполнять ОТТ в рамках одного проекта несколькими исполнителями, а также выполнять проект обогрева на завершающей стадии разработки проектной документации, учитывая все изменения в проекте.

Авторы выражают искреннюю признательность и глубокую благодарность компании АО «Гипрогазоочистка» за возможность использования материально-технической базы предприятия для разработки программного комплекса, директору департамента САПР А.А. Фролову за постановку задачи и руководство разработкой комплекса, инженеру 1-й категории монтажного отдела Р.Р. Салихову за активное участие в тестировании и внедрении программного комплекса проектирования ОТТ тепловыми спутниками для САПР AVEVA PDMS.

Литература

1. TraceCalc Pro. Industrial Pipe and Vessel Heat Trace System Design. URL: <http://www.pentairthermal.com/design-tools/downloadable-tools/trace-calc-pro/index.aspx> (дата обращения: 20.06.2015).
2. Thermon Manufacturing Company | The Heat Tracing Specialist. URL: <http://www.thermon.ru/> (дата обращения: 20.06.2015).
3. ChromaTrace 3.0 – Just Released. URL: <http://www.chromalox.com/resources-and-support/software.aspx#chromatrace> (дата обращения: 20.06.2015).
4. Engineering & Design. URL: <http://www.pentairthermal.com/>

es/services/engineering-and-design/index.aspx (дата обращения: 20.06.2015).

5. Магалиф В.Я., Иткина Д.М., Корельштейн Л.Б. Монтажное проектирование химических, нефтехимических и нефтеперерабатывающих производств. М.: Навигатор, 2010. 344 с.

6. Ellenberger J.P. Piping and Pipeline Calculations Manual Construction, Design Fabrication and Examination. 2nd ed., 2014, 398 p.

7. Nayyar M.L., (ed). Piping handbook. 7th ed., 2482 p.

8. Миркин А.З., Усиныш В.В. Трубопроводные системы: справ. изд. М.: Химия, 1991. 256 с.

9. Хижняков С.В. Практические расчеты тепловой изоляции; [изд. 3-е]. М.: Энергия, 1976. 145 с.

10. TM-1401 AVEVA Plant (12 Series) Programmable Macro Language, AVEVA Solutions Ltd, High Cross, Madingley Road, Cambridge, CB3 0HB, UK, 2009, 115 p.

DOI: 10.15827/0236-235X.112.244-248

Received 23.06.15

THE SOFTWARE DESIGN PACKAGE FOR TECHNOLOGICAL PIPELINES HEATING BY HEAT TRACERS IN COMPUTER-AIDED DESIGN SYSTEMS

Kokhov T.A., Postgraduate Student, tkohov@ggo.ru

(Giprogaзoochistka, Pervomayskaya St. 126, Moscow, 105203, Russian Federation);

Korelshteyn L.B., Ph.D. (Physics and Mathematics), Deputy Director, korelstein@truboprovod.ru

(NTP Truboprovod, Plekhanova St. 7, Moscow, 111141, Russian Federation);

Gartman T.N., Dr.Sc. (Engineering), Professor, Head of Chair, gartman@muctr.ru

(D. Mendeleev University of Chemical Technology of Russian Federation,

Miusskaya Sq. 9, Moscow, 125047, Russian Federation)

Abstract. Industrial plants use the following types of technological pipelines heating: trace heating; steam tracing; electric trace heating. Depending on performance specification or environmental conditions one of these types of heating can dominate in the project.

About 10–15 % of design works (in wiring design) are technological pipelines heating design.

In certain cases almost 40 % of pipelines in the design project should be heated by heat tracers. Sometimes this index can reach 50 %. Thus, about 80 % of heating engineering labor costs is elaboration of heat tracers' traces in a 3D model.

To speed-up engineering of technological pipelines heating by heat tracers the authors developed a software design package for technological pipelines heating implemented in 3D-design AVEVA PDMS.

A distinguishing feature of the developed software design package for technological pipelines heating in comparison with other systems is that it is integrated into AVEVA PDMS. It allows executing modeling of technological pipelines heat tracers in 3D models and releasing necessary target documents: isometric drawings of pipelines.

Software package functionality is verified when developing technological pipelines water heating by heat tracers of sulfur recovery.

Keywords: heat tracing, power supply, reversing, pipe manifold.

References

1. *TraceCalc Pro. Industrial Pipe and Vessel Heat Trace System Design*. Available at: <http://www.pentairthermal.com/design-tools/downloadable-tools/trace-calc-pro/index.aspx> (accessed June 20, 2015).

2. *Thermon Manufacturing Company | The Heat Tracing Specialist*. Available at: <http://www.thermon.ru/> (accessed June 20, 2015).

3. *ChromaTrace 3.0 – Just Released*. Available at: <http://www.chromalox.com/resources-and-support/software.aspx#chromatrace> (accessed June 20, 2015)

4. *Engineering & Design*. Available at: <http://www.pentairthermal.es/services/engineering-and-design/index.aspx> (accessed June 20, 2015).

5. Magalif V.Ya., Itkina D.M., Korelshteyn L.B. *Montazhnoe proektirovanie khimicheskikh, neftekhimicheskikh i neftepererabatyvayushchikh proizvodstv* [Installation design of chemical, petrochemical and refining industries]. Moscow, Navigator Publ., 2010, 344 p.

6. Ellenberger J.P. *Piping and Pipeline Calculations Manual Construction. Design Fabrication and Examination*. 2nd ed., 2014, 398 p.

7. Nayyar M.L. *Piping handbook*. 7th ed., 2482 p.

8. Mirkin A.Z., Usinsh V.V. *Truboprovodnye sistemy: sprav. izd.* [Pipeline Systems: Ref. Ed.]. Moscow, Khimiya Publ., 1991, 256 p. (in Russ).

9. Khizhnyakov S.V. *Prakticheskie raschety teplovoy izolyatsii* [Practical Calculations of Thermal Insulation]. 3rd ed., Moscow, Energiya Publ., 1976, 145 p.

10. *TM-1401 AVEVA Plant (12 Series) Programmable Macro Language*. AVEVA Solutions Ltd, High Cross, Madingley Road, Cambridge, CB3 0HB, UK, 2009, 115 p.

УДК 66.011, 60.7

Дата подачи статьи: 09.09.15

DOI: 10.15827/0236-235X.112.249-255

ПОДХОД К МОДЕЛИРОВАНИЮ, МАСШТАБИРОВАНИЮ И ОПТИМИЗАЦИИ РАБОТЫ БИОРЕАКТОРОВ НА ОСНОВАНИИ ВЫЧИСЛИТЕЛЬНОЙ ГИДРОДИНАМИКИ

Е.В. Гусева, к.т.н., доцент, eguseva@rally-online.ru;

Р.Р. Сафаров, соискатель;

Н.В. Меньшуткина, д.т.н., профессор

*(Российский химико-технологический университет им. Д.И. Менделеева,
Миусская пл., 9, г. Москва, 125047, Россия);*

Ж. Бударн, профессор

*(Университет Лоррен, лаборатория реакций и химической инженерии, LRGP, ENSAIA,
авеню Форе де айе 2, 40602, 54518 Вандевр-ле-Нанси)*

За последние 20 лет вычислительная гидродинамика зарекомендовала себя как инструмент для решения задач, включающих одно или более из следующих явлений: движение потоков жидкости, теплопередача, массоперенос, химическая реакция. В данной работе CFD был использован для реализации предложенного обобщенного подхода к моделированию, масштабированию и оптимизации биореакторов, один из этапов которого заключается в использовании современного метода цифровой трассерной визуализации PIV, позволяющего определять скорости частиц по их изображениям. В качестве примера рассмотрен процесс моделирования, расчета и оптимизации получения молочной кислоты бактериями *Lactobacillus casei* в реакторе смешения периодического действия. При расчете учитывалось влияние одного из аспектов механического стресса, перемешивания, на жизнедеятельность и продуктивность бактерий. Максимальная скорость диссипации энергии выбрана в качестве характеристики механического стресса. Были определены размеры турбулентных характеристических вихрей в биореакторе, негативно влияющих на жизнедеятельность клеток/бактерий. Введено понятие удельного объема стрессовой зоны. На его основании была определена предельная скорость перемешивания в данном биореакторе смешения, превышение которой ведет к значительному воздействию механического стресса на бактерии. Данный подход может быть использован для перехода к большим реакторам и поиску оптимальных условий работы.

Ключевые слова: моделирование, масштабирование, оптимизация, CFD, биореактор, микроорганизмы, клетки/бактерии, механический стресс, скорость диссипации энергии, стрессовая зона.

Вычислительная гидродинамика (computational fluid dynamics, CFD) широко используется для решения различных задач. Среди наиболее известных программных пакетов, работающих с CFD, стоит назвать FLUENT, CFX, FLOWIZARD, PHOENICS, STAR-CD, OPEN FOAM [1, 2].

Вычислительная гидродинамика представляет собой подход, который использует численные методы для решения задач, связанных с описанием движения потоков жидкости. Это осуществляется путем разбиения жидкости на мелкие ячейки для создания расчетной области, называемой сеткой, а затем применения к ним подходящего алгоритма для решения различных уравнений движения [3]. Дифференциальные уравнения Навье–Стокса, содержащие законы сохранения массы, импульса и энергии, являются основой CFD и решаются с использованием таких методов дискретизации, как метод конечных разностей, метод конечных элементов в сочетании с численными методами (например метод Particle-in-Cell (PIC – метод частиц в ячейках) для решения задач гидродинамики с одной, двумя и тремя пространственными переменными), метод Marker-and-Cell (MAC – метод маркеров и ячеек), произвольные методы Лагранжа–Эйлера и модели *k-ε*-турбулентности и так далее [1, 3]. Решение осуществляется для каждой

ячейки, в результате чего формируется поле течения.

Моделирование в CFD-пакетах обычно включает три этапа. На первом этапе строится геометрическая модель объекта с использованием соответствующего разбиения на ячейки, граничных условий и информации о свойствах потока. На втором этапе решаются уравнения сохранения импульса, массы и энергии и получаются результаты анализа. На третьем этапе происходят интерпретация данных в виде векторов и контурных графиков и их последующий анализ для определения необходимого решения.

Применение CFD для сложных и многофакторных процессов, протекающих в биореакторах, позволяет глубже понять процесс, что дает возможность (например, путем постановки вычислительного эксперимента) значительно снизить общее количество требуемых экспериментальных исследований и в итоге получить продукт нужного качества с меньшими затратами. В настоящее время CFD широко используется в таких областях биотехнологии, как процессы ферментации, центрифугирование, хроматография, ультрафильтрация, микрофильтрация и сублимационная сушка [4–6].

В числе самых распространенных областей применения CFD в биотехнологии стоит назвать

моделирование, расчет, оптимизацию и масштабирование биореакторов различного типа, в первую очередь реакторов смешения.

Обобщенный подход к моделированию, масштабированию и оптимизации биореакторов

Процессы, протекающие в биореакторах с участием микроорганизмов или живых клеток (клетки животных, растений или человека), отличаются высокой сложностью. Необходимо учитывать поступление питательных компонентов, высвобождение различных метаболитов, представляющих собой основной продукт, побочные продукты или отходы жизнедеятельности клеток. Перемешивание и аэрация вносят основной вклад в создание гидродинамической обстановки в биореакторе, что напрямую влияет на доставку питательных веществ к клетке и удаление метаболитов, а также на их жизнедеятельность. Более того, влияние различных факторов на течение процесса усложняет задачу оптимизации и масштабирования.

Разработанный обобщенный подход к моделированию, масштабированию и оптимизации биореакторов представлен на рисунке 1.

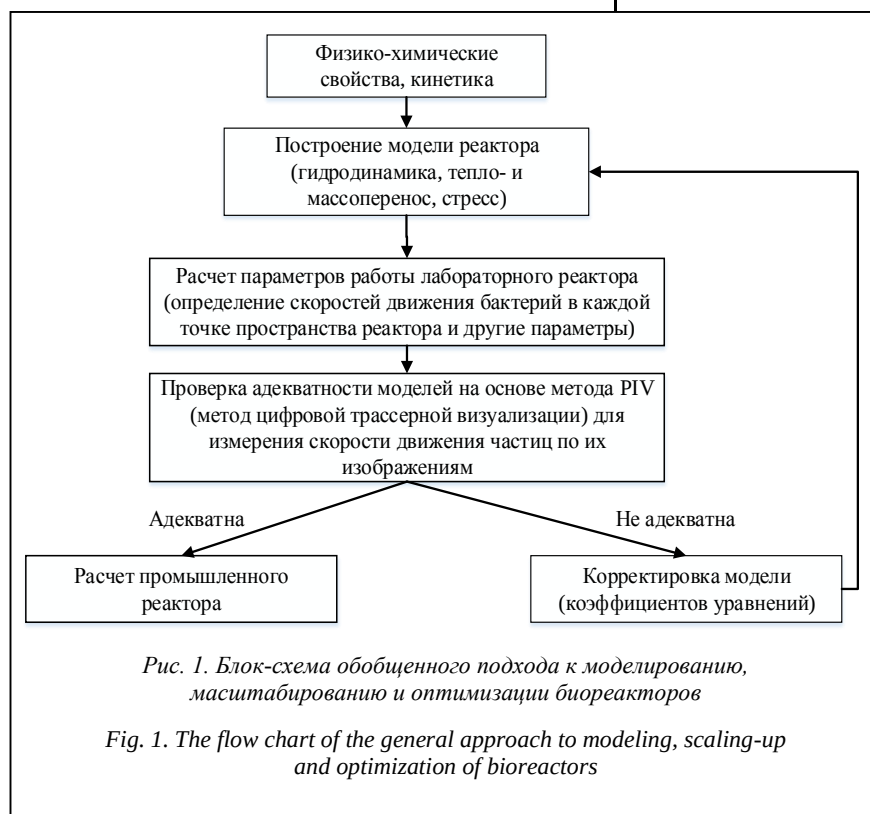
лученных данных составляется математическая модель биореактора, учитывающая процессы гидродинамики, тепло- и массопереноса и явления стресса. На следующем этапе рассчитываются параметры работы лабораторного реактора, в том числе определяются скорости движения клеток/бактерий в каждой точке пространства реактора. В качестве подхода для определения адекватности разработанной модели и используемого для масштабирования реактора был предложен метод PIV (Particle Image Velocimetry), метод цифровой трассерной визуализации, позволяющий измерять векторные поля скорости в выбранном сечении потока газа или жидкости. Одним из достоинств данного метода является возможность измерения в широком динамическом диапазоне скоростей (порядка 500:1), что делает его весьма привлекательным для исследования сложных турбулентных потоков. В зависимости от результата проверки достоверности математического моделирования происходит либо переход к расчету реактора промышленных размеров, либо корректировка уравнений модели и возврат на стадию моделирования.

Частичная реализация данного обобщенного подхода была рассмотрена на примере моделирования, расчета и оптимизации процесса получения молочной кислоты (одной из органических кислот, получаемых биотехнологическим путем) в биореакторе периодического действия при учете воздействия механического стресса на бактерии *Lactobacillus casei* (далее частицы) с использованием средств вычислительной гидродинамики CFD.

Экспериментальные исследования были проведены на основе разработанного *полного факторного эксперимента* (ПФЭ) для двух факторов при их варьировании на трех уровнях. В качестве факторов были взяты влияние скорости перемешивания как аспект механических стрессовых воздействий и начальная концентрация субстрата (питательная среда) как аспект химических стрессовых воздействий.

При моделировании биореактора были сделаны следующие допущения:

- все бактерии (частицы) имеют сферическую форму и один диаметр (принято 1 мкм);



Сначала анализируются физико-химические параметры культуры клеток/бактерий исследуемого процесса, на основе чего моделируются кинетические зависимости роста клеток, продукта и расхода питательных веществ. Затем с использованием по-

- вероятность столкновения частиц в объеме аппарата равна 0;
- все столкновения частиц со стенками аппарата являются абсолютно упругими, то есть потери энергии отсутствуют;
- в связи с интенсивным перемешиванием достигается равномерное распределение частиц по всему объему биореактора;
- плотность, вязкость и коэффициент диффузии культуральной среды постоянны и не зависят от времени и координат в биореакторе;
- скорость подачи воздуха для аэрации в биореакторе незначительна, в связи с этим считаем, что основное воздействие механического (в данном случае гидродинамического) стресса происходит в результате перемешивания четырехлопастной мешалкой.

Создание геометрии биореактора и расчетной сетки

На первом этапе моделирования была построена геометрическая модель периодического биореактора смешения. Геометрия емкостного биореактора создана в Gambit, программе-генераторе расчетных сеток, включенной в пакет FLUENT 6.1.22. Все размеры соответствуют аналогичным размерам в реальном биореакторе.

Существует несколько разновидностей расчетных сеток в объемах, самые распространенные из них – гексагональная и тетраэдрическая. На данном этапе была выбрана тетраэдрическая разновидность для большей изотропности рабочей сетки (рис. 2).

Основной параметр расчетной сетки при построении – ее плотность. Чем больше ячеек в сетке, тем меньше ошибка в каждой ячейке, а следовательно, и общая ошибка модели. С другой стороны, плотность расчетной сетки влияет на время сходимости решения, и при большом количестве ячеек для сходимости могут потребоваться значительные временные ресурсы, что зачастую неприемлемо при решении сложных задач. Из-за этого приходится прибегать к различным способам уменьшения числа ячеек, основным из которых является снижение плотности сетки в областях, где поведение потоков в достаточной мере прогнозируемо или где можно пренебречь отклонениями.

Выбор размерности сетки и ее плотности зависит исключительно от опыта и знаний человека, решающего задачу моделирования, а также от аппаратного оформления компьютера, на котором осуществляется моделирование. При необходимости этот параметр корректируется путем проведения и анализа пробных запусков расчета. Оптимальные параметры построения расчетной сетки подбираются интерактивно.

Поскольку биореактор с перемешивающим устройством представляет собой аппарат со сложной структурой потоков, особое внимание при построении расчетной сетки было уделено мешалке: в связи с наличием мелких деталей плотность сетки на ее поверхности была увеличена. В свою очередь, плотность сетки на поверхности аппарата была снижена для обеспечения приемлемого времени сходимости расчета. Это было реализовано путем задания специальной функции по рабочему объему биореактора. Параметры функции – минимальный

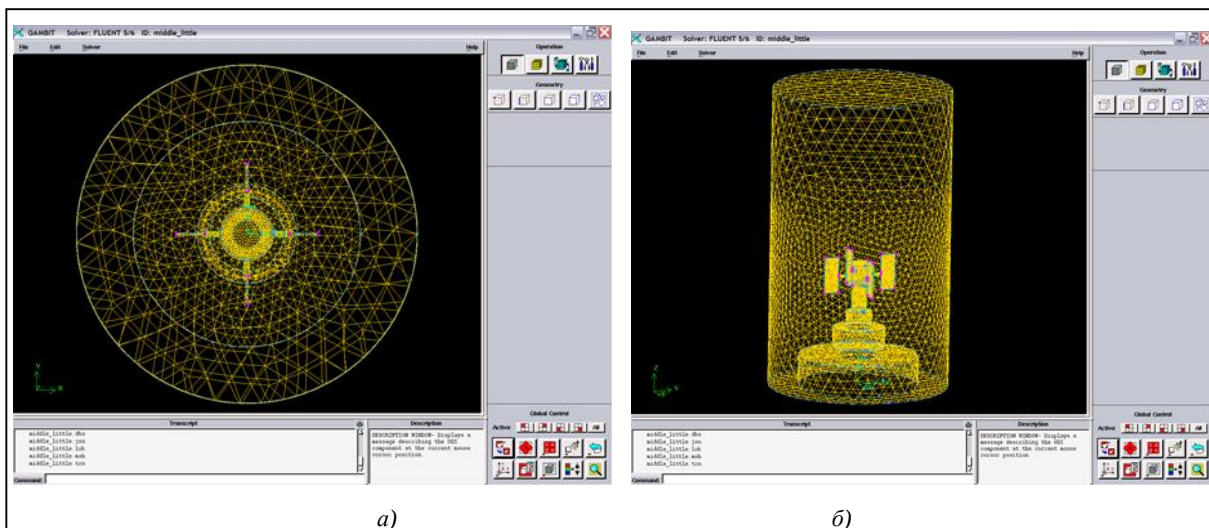


Рис. 2. Расчетная сетка по рабочему объему биореактора:
а) биореактор с расчетной сеткой, вид сверху; б) расчетная сетка биореактора

Fig. 2. The computation grid of bioreactor's working volume:
a) the bioreactor with the computational grid, top view; б) the bioreactor's computational grid

и максимальный размеры ячеек, скорость увеличения размеров ячеек. В результате выбрана расчетная сетка, представленная на рисунке 2, где параметры функции размера имеют следующие значения: минимальный размер ячеек – 1 мм, максимальный – 10 мм, скорость роста – 1,1. Число ячеек в объеме биореактора составило около 110 000.

Моделирование биореактора смешения и результаты расчетов

Для данного аппарата были заданы следующие граничные условия для соответствующих областей: Wall 1 (поверхность мешалки) является стационарной относительно неподвижной системы отсчета, Wall 2 (стенка аппарата) движется вместе с рабочим объемом в биореакторе, коэффициент шероховатости обеих областей равен 0,5.

На основании допущения, что все столкновения частиц абсолютно упруги, для данных граничных условий был выбран тип взаимодействия частиц диспергированной фазы со стенками аппарата – «отражение». Математическая запись граничного условия:

$$\vec{v}(x_{cm}, y_{cm}, z_{cm}) = 0. \quad (1)$$

Вращающаяся система отсчета (Rotating Reference Frame) была использована для моделирования вращения мешалки. Мешалка будет иметь нулевую скорость, скорость вращения рабочего объема вокруг мешалки равна постоянной скорости вращения мешалки в реальном аппарате.

Оценка влияния механического стресса в биореакторе проводилась на основе скорости диссипации энергии ε [7]. Выражение для скорости диссипации энергии имеет вид

$$\varepsilon = \frac{1}{2} \nu \cdot \sum_{i,j} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)^2, \quad u'_i = u_i - \bar{u}_i, \quad (2)$$

где u_i – реальная скорость; $\bar{u}_i$ – средняя скорость; u'_i – пульсационная скорость.

На основе теории изотропной турбулентности, разработанной А.Н. Колмогоровым, турбулентность объясняется наличием большого количества вихрей. Первичные неустойчивые вихри большого размера, разрушаясь, образуют вихри меньшего размера, при этом имеет место диссипация энергии. Размер вихрей определяет так называемый масштаб турбулентности [7–9]. Для нахождения реакции микроорганизмов на гидродинамический стресс необходимо сравнить их размер с характеристическим размером турбулентных вихрей:

$$l = \left(\frac{\nu^3}{\varepsilon} \right)^{1/4}, \quad (3)$$

где ν – кинематическая вязкость.

В пакете Fluent для математического описания двухфазных течений чаще всего используют два метода: дискретно-траекторный Эйлер–Лагранжа

(Euler–Lagrange) и многожидкостной Эйлер–Эйлера (Euler–Euler). В первом методе рассчитываются траектории и характеристики индивидуальных частиц в определенные промежутки времени, то есть для частиц используется метод Лагранжа, а для жидкой фазы метод Эйлера. Во втором методе дисперсная фаза (частицы) рассматривается как сплошная среда (континуум), а общее течение – как смесь двух или более жидкостей [10].

Для данного процесса объемное содержание дисперсной фазы (бактерий) в относительно малом объеме потока не превышает 10–12 %, внутренний объем биореактора представляет собой сплошную среду. Такая система рассматривается как однофазная, поток является изотропным, поэтому для дисперсной фазы (бактерий) был применен метод Эйлер–Эйлера. Для многофазного потока была применена многофазная модель Эйлера, решающая совместно уравнение сохранения массы:

$$\frac{\partial \alpha_1}{\partial t} + \nabla(\alpha_1 \vec{v}_1) = 0, \quad (4)$$

$$\frac{\partial \alpha_2}{\partial t} + \nabla(\alpha_2 \vec{v}_2) = \frac{-\alpha_2}{\rho_2} \cdot \frac{d_2 \rho_2}{dt}, \quad (5)$$

и уравнения движения:

$$\frac{\partial}{\partial t} (\alpha_1 \rho_1 \vec{v}_1 + \nabla(\alpha_1 \rho_1 \vec{v}_1 \vec{v}_1)) = \quad (6)$$

$$= -\alpha_1 \nabla p + \nabla \bar{\tau}_1 + \alpha_1 \rho_1 \vec{g} + \alpha_1 \rho_1 \vec{F},$$

$$\frac{\partial}{\partial t} (\alpha_2 \rho_2 \vec{v}_2 + \nabla(\alpha_2 \rho_2 \vec{v}_2 \vec{v}_2)) = \quad (7)$$

$$= -\alpha_2 \nabla p + \nabla p_s + \nabla \alpha_2 \rho_2 \vec{g} + \alpha_2 \rho_2 \vec{F} + \sum_{l=1}^2 K_{12} (\vec{v}_1 - \vec{v}_2).$$

для каждой из фаз.

Индексы 1 и 2 обозначают сплошную (культуральная среда) и дисперсную (бактерии) фазы соответственно; t – время (с); u_q – вектор скоростей q -й фазы (м/с); p – давление в биореакторе (Па); ρ_q – плотность q -й фазы (кг/м³); g – ускорение свободного падения (м/с²); p_s – давление твердой фазы (Па); F – внешние силы (подъемная сила и др.) (Н); K_{pq} – коэффициент, характеризующий обмен импульсом между фазами p и q , определяется при помощи эмпирических соотношений; α_q – объемная доля q -й фазы; τ – тензор напряжений q -й фазы (Па); μ_q – динамическая вязкость q -й фазы (Па·с); λ_q – объемная вязкость q -й фазы, характеризует деформацию объемного сжатия (Па·с).

Для турбулентного режима движения в биореакторе (рассчитанные значения критерия Рейнольдса составили от 3 000 до 15 000 в зависимости от скорости вращения мешалки) во Fluent была выбрана модель k - ε как одна из наиболее распространенных моделей турбулентности, хорошо отображающих потоки, скорость которых на порядок ниже скорости звука. Альтернативная модель k - ω с большей точностью описывает пристеночные эф-

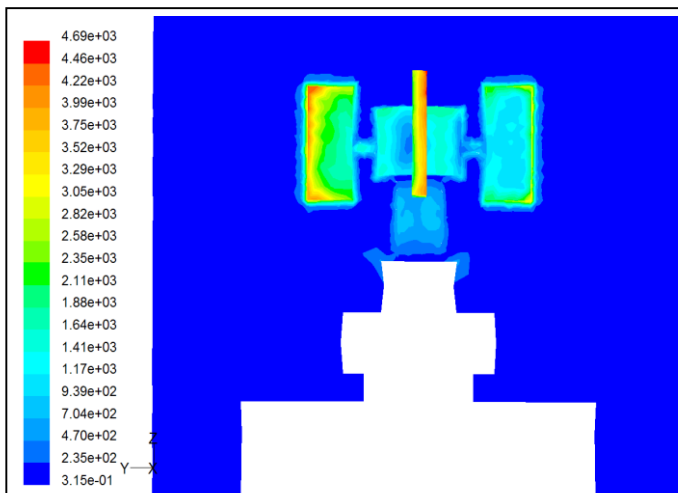


Рис. 3. Профиль скоростей диссипации энергии по биореактору
(скорость перемешивания 1 000 об./мин.)

Fig. 3. The profile of energy dissipation rates (agitation rate 1000 rpm)

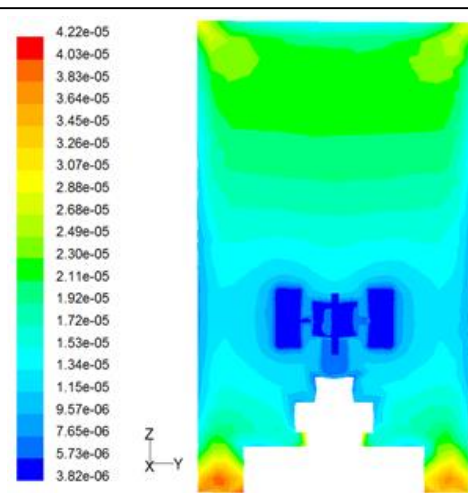


Рис. 4. Профиль характеристических размеров турбулентных вихрей по объему биореактора
(1 000 об./мин.)

Fig. 4. The profile of characteristic sizes of bioreactor volume turbulence eddies (agitation rate 1000 rpm)

факты, но сходимость по сравнению с выбранной моделью k - ε хуже. Кроме того, модель k - ε позволяет вычислять скорости диссипации энергии по аппарату, таким образом показывая уровень механического стресса в аппарате.

Выбранная модель k - ε в применении к двухфазной системе может представляться различными уравнениями. При моделировании использовалась дисперсная модель турбулентности, которую целесообразно использовать при небольших объемах дисперсных фаз. Данная модель представляется следующими уравнениями:

$$\frac{\partial}{\partial t}(\alpha_q \rho_q \varepsilon_q) + \nabla \cdot (\alpha_q \rho_q \varepsilon_q \vec{U}_q) = \nabla \cdot \left(\alpha_q \frac{\mu_{t,q}}{\sigma_\varepsilon} \nabla \varepsilon_q \right) +$$

$$+ \alpha_q \frac{\varepsilon_q}{k_q} (C_{1\varepsilon} G_{k,q} - C_{2\varepsilon} \rho_q \varepsilon_q) + \alpha_q \rho_q \Pi_{\varepsilon_q},$$

$$\frac{\partial}{\partial t}(\alpha_q \rho_q k_q) + \nabla \cdot (\alpha_q \rho_q k_q \vec{U}_q) = \nabla \cdot \left(\alpha_q \frac{\mu_{t,q}}{\sigma_k} \nabla k_q \right) +$$

$$+ \alpha_q G_{k,q} - \alpha_q \rho_q \varepsilon_q + \alpha_q \rho_q \Pi_{k_q},$$

где k_q – кинетическая энергия турбулентности q -й фазы, $\text{м}^2/\text{с}^2$; ε_q – скорость диссипации кинетической энергии q -й фазы, $\text{м}^2/\text{с}^3$; $G_{k,q}$ – производство кинетической энергии q -й фазы, вызванное градиентом скорости; $\vec{U}_q$ – вектор средневзвешенных скоростей q -й фазы; $\Pi_{k_q}, \Pi_{\varepsilon_q}$ – влияние дисперсной фазы на сплошную фазу q ; σ_k и σ_ε – числа Прандтля соответственно для k и ε ; $C_{1\varepsilon}, C_{2\varepsilon}$ – эмпирические константы, полученные из опытов с воздухом и водой и являющиеся универсальными для большого числа задач гидродинамики и газодинамики. Значения констант: $C_{1\varepsilon} = 1,44$, $C_{2\varepsilon} = 1,92$.

Граничные условия для данных уравнений:

$$k_q(x_0; y_0; z_0) = k_{q,0}, \quad (10)$$

$$\varepsilon_q(x_0; y_0; z_0) = \varepsilon_{q,0}. \quad (11)$$

Результаты и обсуждение

На основании расчетов были получены профили скоростей диссипации энергии по аппарату. Подтверждено, что наибольшие значения скорости диссипации энергии, выделенные красным и желтым цветом, наблюдаются около мешалки (рис. 3). Кроме того, определены профили характеристических размеров турбулентных вихрей по объему биореактора при различных скоростях вращения мешалки (рис. 4). В качестве примера даны графики для скорости перемешивания 1 000 об./мин. Наименьшие значения характеристических размеров вихрей, оказывающие разрушительное действие на клетки, соответствуют области вблизи мешалки.

Было определено, что бактерии подвергаются воздействию механического стресса при размерах турбулентных вихрей, меньших 10 мкм (что соответствует размерам клеток, 0,1–10 мкм). Зона, где наблюдается стрессовое воздействие, называется стрессовой зоной [11]. Увеличение скорости вращения мешалки приводит к возрастанию объема стрессовой зоны в биореакторе (рис. 5). Было введено понятие удельного объема стрессовой зоны к рабочему объему биореактора. На основании полученных результатов построена зависимость удельного объема стрессовой зоны биореактора от скорости вращения мешалки (рис. 6). Анализ полученной зависимости показывает, что при

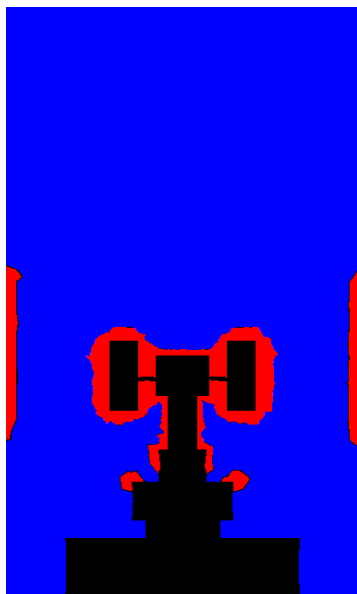


Рис. 5. Стрессовая зона в биореакторе (красный цвет) (1 000 об./мин.)

Fig. 5. Bioreactor's stress zone (red colour, 1000 rpm)

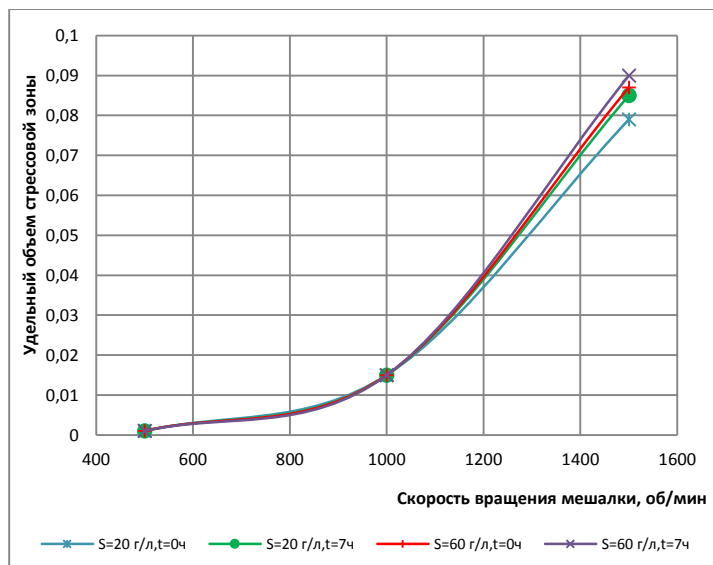


Рис. 6. Зависимости удельного объема стрессовой зоны в биореакторе от скорости перемешивания при различных начальных концентрациях субстрата и биомассы

Fig. 6. Relations between stress zone specific volume and agitation rates when substrate and biomass initial concentrations are different

достижении примерно 700 об./мин. начинается резкое возрастание удельного объема стрессовой зоны и, соответственно, самого объема этой зоны. Таким образом, будет наблюдаться значительное увеличение воздействия механического стресса на клетки в биореакторе. Превышение данного значения скорости перемешивания приводит к снижению продуктивности по основному продукту – молочной кислоте.

Таким образом, был предложен обобщенный подход к моделированию, масштабированию и оптимизации биореакторов, один из этапов которого заключается в использовании современного метода цифровой трассерной визуализации PIV, позволяющий определять скорости частиц по их изображениям. Частичная реализация данного подхода была рассмотрена на примере моделирования, расчета и оптимизации процесса получения молочной кислоты бактериями *Lactobacillus casei* в реакторе смешения периодического действия с использованием программного пакета CFD. При расчете учитывалось влияние одного из аспектов механического стресса, перемешивания, на жизнедеятельность и продуктивность бактерий. В качестве характеристики механического стресса выбрана максимальная скорость диссипации энергии. Было введено понятие удельного объема стрессовой зоны, на основании которого определена предельная скорость перемешивания в данном биореакторе смешения, превышение которой ведет к значительному воздействию механического стресса на бактерии. Данный подход может быть использован для перехода

к большим реакторам и поиску оптимальных условий работы.

Литература

1. Sharma C., Malhotra D., Rathore A.S. Review of computational fluid dynamics applications in biotechnology processes. *Biotechnol. Prog.*, 2011. vol. 27, no. 6, pp. 1497–1510.
2. Huttmacher D.W., Singh H. Computational fluid dynamics for improved bioreactor design and 3D culture. *Trends in Biotechnol.*, 2008, vol. 26, no. 4, pp. 166–172.
3. CFD online. URL: <http://www.cfd-online.com> (дата обращения: 06.09.2015).
4. Kaiser S.C., Löffelholz C., Werner S., Eibl D. CFD for characterizing standard and single-use stirred cell culture bioreactors, in: Minin I. (Eds.), *Intech*, 2011, pp. 97–122.
5. Johnson C., Natarajan M., Antoniou C. Verification of energy dissipation rate scalability in pilot and production scale bioreactors using computational fluid dynamics. *Biotechnol. Progr.*, 2014, vol. 30, no. 6, pp. 760–764.
6. Singh H., Huttmacher DW. Bioreactor studies and computational fluid dynamics. *Adv Biochem Eng Biotechnol*, 2009, vol. 112, pp. 231–249.
7. Henzler H.-J. Partial stress in bioreactors. *Adv. Biochem. Eng. / Biotechnol.*, 2000, vol. 67, pp. 35–82.
8. Колмогоров А.Н. Локальная структура турбулентности в несжимаемой жидкости при очень больших числах Рейнольдса // Докл. АН СССР. Т. 30. № 4. С. 299–303.
9. Hu W., Berdugo C., Chalmers J.J. The potential of hydrodynamic damage to animal cells of industrial relevance: current understanding. *Cytotechnol.*, 2011, vol. 63, pp. 445–460.
10. William J. Kelly. Using computational fluid dynamics to characterize and improve bioreactor performance. *Biotechnol. Appl. Biochem.*, 2008, vol. 49, pp. 225–238.
11. Boudrant J., Menshutina N.V., Skorokhodov A.V., Guseva E.V., Fick M. Mathematical modelling of cell suspension in high cell density conditions. Application to L-lactic acid fermentation using *Lactobacillus casei* in membrane bioreactor. *Process Biochem.* 2005, vol. 40, no. 5, pp. 1641–1647.

DOI: 10.15827/0236-235X.112.249-255

Received 09.09.15

**THE APPROACH TO MODELING, SCALING-UP AND OPTIMIZATION OF BIOREACTORS
BASED ON COMPUTATIONAL FLUID DYNAMICS****Guseva E.V.**, PhD. (Engineering), Associate Professor, eguseva@rally-online.ru;**Safarov R.R.**, Candidate;**Menshutina N.V.**, Dr.Sc. (Engineering), Professor(D. Mendeleev University of Chemical Technology of Russian Federation,
Miusskaya Sq. 9, Moscow, 125047, Russian Federation);**Boudrant J.**, Professor(Universite de Lorraine, Laboratory Reaction and Process Engineering,
LRGP, ENSAIA, 2 Avenue de la Forêt de Haye, TSA 40602, 54518 Vandoeuvre-lès-Nancy)

Abstract. During last 20 years computational fluid dynamics (CFD) proved itself as an instrument to solve problems with one or more phenomena, such as liquid flow, heat and mass transfer, chemical reaction. This article is devoted to the general approach to modeling, scaling-up and optimization of bioreactors using CFD. One of the steps the proposed approach is based on PIV (particle image velocimetry), which helps to determine particles velocities from their images. As an example the paper considers the process of modeling, scaling-up and optimization of lactic acid production by *Lactobacillus casei* bacteria in a stirred-tank batch bioreactor. The mechanical stress aspect (agitation rate) which influences on bacteria viability and productivity has been taken into account during the calculation. The maximum dissipation rate has been selected as a characteristic of mechanical stress. The authors defined the sizes of turbulence characteristic eddies in bioreactor that negatively affected bacteria. The authors have introduced the concept of stress zone specific volume, which has been used to define the limiting agitation rate in this bioreactor. The approach can be used for scaling-up and searching of optimal operation conditions.

Keywords: modeling, scaling-up, optimization, CFD, bioreactor, microorganisms, cells/bacteria, mechanical stress, energy dissipation rate, stress zone.

References

1. Sharma C., Malhotra D., Rathore A.S. Review of Computational Fluid Dynamics Applications in Biotechnology Processes. *Biotechnol. Prog.* 2011, vol. 27, no. 6, pp. 1497–1510.
2. Hutmacher D.W., Singh H. Computational fluid dynamics for improved bioreactor design and 3D culture. *Trends in Biotechn.* 2008, vol. 26, no. 4, pp. 166–172.
3. *CFD online*. Available at: <http://www.cfd-online.com/> (accessed September 6, 2015).
4. Kaiser S.C., Löffelholz C., Werner S., Eibl D. *CFD for Characterizing Standard and Single-use Stirred Cell Culture Bioreactors*. Minin I. (Eds.), Intech, 2011. pp. 97–122.
5. Johnson C., Natarajan M., Antoniou C. Verification of energy dissipation rate scalability in pilot and production scale bioreactors using computational fluid dynamics. *Biotechnol. Progr.* 2014, vol. 30, no. 6, pp. 760–764.
6. Singh H., Hutmacher DW. Bioreactor studies and computational fluid dynamics. *Adv. Biochem. Eng. Biotechnol.* 2009, vol. 112, pp. 231–249.
7. Henzler H.-J. Partical stress in bioreactors. *Adv. Biochem. Eng. Biotechnol.* 2000, vol. 67, pp. 35–82.
8. Kolmogorov A.N. Local turbulence structure in incompressible liquid with Reynolds big numbers. *Dokl. Akad. nauk. Ser. Fiz.* [Reports of USSR Academy of Science]. vol. 30, no. 4, pp. 299–303 (in Russ.).
9. Hu W., Berdugo C., Chalmers J.J. The potential of hydrodynamic damage to animal cells of industrial relevance: current understanding. *Cytotechn.* 2011, vol. 63, pp. 445–460.
10. Kelly W.J. Using computational fluid dynamics to characterize and improve bioreactor performance. *Biotechnol. Appl. Biochem.* 2008, vol. 49, pp. 225–238.
11. Boudrant J., Menshutina N.V., Skorokhodov A.V., Guseva E.V., Fick M. Mathematical modelling of cell suspension in high cell density conditions. Application to L-lactic acid fermentation using *Lactobacillus casei* in membrane bioreactor. *Process Biochem.* 2005, vol. 40, no. 5, pp. 1641–1647.

УДК 004.054

Дата подачи статьи: 16.09.15

DOI: 10.15827/0236-235X.112.256-260

ПРОГРАММА ПО ОЦЕНКЕ КАЧЕСТВА АКАДЕМИЧЕСКИХ ПРОГРАММНЫХ ПРОДУКТОВ НА ОСНОВЕ МЕТОДИКИ ХОЛСТЕДА

(Работа выполнена при частичной поддержке РФФИ, грант № 14-07-00665)

*А.И. Афанасова, аспирант, младший научный сотрудник, nastyaaafanasova@yandex.ru
(Ульяновский государственный университет, ул. Льва Толстого, 42, г. Ульяновск, 432017, Россия)*

Оценивание академических программных продуктов заметно сказывается на уровне усвоения сложных дисциплин информационных и вычислительных технологий. Однако в литературе по программометрии практически не делается различий между академическими и профессиональными программными продуктами. Проекты академических программных продуктов в отличие от профессиональных не имеют непосредственной обратной связи по качеству с конкурентной рыночной средой. Учитывая большой объем документации, фронтальное оценивание академических программных продуктов становится трудоемким делом, часто не дающим нужного эффекта – быстрого и/или заметного прироста компетенций студента ввиду неизбежной субъективности – опоры на личный опыт, знания и представления преподавателя. Решение видится в создании специальных методик, отличных от методик оценивания профессиональных программных продуктов, и в разработке инструментария преподавателя – специализированных программ оценивания академических программных продуктов. Ожидается, что эти инструменты, благодаря создаваемой ими объективной обратной связи, будут содействовать модификации поведения студентов, что выразится в повышении качества не только самих академических программных продуктов, но и всего проектно-ориентированного образовательного процесса.

В статье представлены результаты разработки необходимого инструмента оценивания академических программных продуктов. С использованием методики Холстеда построен вычислительный алгоритм формирования количественных показателей качества таких продуктов, создаваемых студентами на языке программирования C++. На языке C# разработана оригинальная программа оценки качества академических программных продуктов на основе методики Холстеда. Опытной эксплуатацией разработанной программы подтверждены ожидаемые результаты.

Ключевые слова: оценивание качества программных продуктов, академические программные продукты, программометрия, метрики Холстеда, проектно-ориентированное обучение, C++.

В современном мире оценивание качества ПО считается одной из важнейших задач как в софтверных компаниях, так и в высших учебных заведениях с компьютерно-ориентированным уклоном подготовки. В таких вузах вопрос о том, как оценивать качество создаваемых студентами программ – так называемых *академических программных продуктов* (АПП), остается открытым, хотя его решение опосредованно затрагивает интересы и софтверных компаний, куда придут работать выпускники. Ввиду наличия ряда отличительных особенностей АПП от *профессиональных программных продуктов* (ППП) оценивание их качества требует специального подхода и особой вычислительной методики.

Оцениванию качества программных продуктов посвящено множество исследований [1–5], однако их объектом, как правило, являются ППП. Оцениванию качества АПП уделяется заметно меньшее внимание, хотя эта задача неизбежно возникает на технических специальностях, где студенты в ходе учебного процесса создают большое число программ [5–8]. В силу особенностей АПП для обработки таких больших массивов проектной документации требуются специальные средства.

Отличительной особенностью АПП от ППП является отсутствие обратной связи по их качеству, поскольку АПП в большинстве случаев не имеют непосредственного контакта с конкурентной рыночной средой. Оценивание качества студенческих

программ обычно выполняется экспертами – преподавателями, которые опираются на свой личный опыт, знания и представления. Однако даже при условии высокой квалификации преподавателей их требования индивидуальны и субъективны. Учитывая то, что во время обучения создается большое количество АПП, их оценивание – процесс достаточно сложный и трудоемкий. В литературе наблюдается явный недостаток сведений о средствах автоматизации этого процесса.

Данная работа предпринята с целью восполнения отмеченного недостатка. В Ульяновском государственном университете при чтении дисциплин, связанных с вычислительной математикой и программированием, с 1989 года применяется проектно-ориентированное обучение [8, 9], которое предусматривает разработку студентами целостных, нетривиальных программных продуктов. Оценив уровень качества по множеству АПП, преподаватель и кафедра получают необходимую обратную связь и, следовательно, возможность сформировать оценку качества всего образовательного процесса [10].

Методика оценивания качества АПП

В основу предлагаемой методики положена работа М. Холстеда [11], применение которой для оценивания качества АПП впервые предложено в работе [10].

В соответствии с методикой Холстеда, оценку качества программного продукта формируют в результате определения специальных характеристик (таких как число операторов и операндов) исходного кода программы. При этом априори считают, что текст программы свободен от ошибок, связанных с языком программирования. Для АПП это справедливо, поскольку для оценивания отбираются тексты программ, успешно прошедшие компиляцию. Также считают, что текст АПП свободен от логических (алгоритмических) ошибок и отклонений от выданного задания, поскольку он уже прошел необходимую проверку у эксперта-преподавателя.

Процесс получения качественных характеристик АПП. По методике Холстеда [11], принципиальным параметром качества АПП является критерий Hq совершенства логической структуры текста программного продукта. Критерий Hq вычисляют с использованием четырех характеристик длины программного текста: η_1 – число простых (или отдельных) операторов, появляющихся в данной реализации (словарь операторов); η_2 – число простых (или отдельных) операндов, появляющихся в данной реализации (словарь операндов); N_1 – общее число всех операторов, появляющихся в данной реализации; N_2 – общее число всех операндов, появляющихся в данной реализации.

На основе этих параметров для каждого текста АПП вычисляют две характеристики длины: экспериментально определенную длину

$$N = N_1 + N_2 \quad (1)$$

и теоретическую длину

$$\hat{N} = \eta_1 * \log_2 \eta_1 + \eta_2 * \log_2 \eta_2. \quad (2)$$

Обоснование корректности формулы вычисления теоретической длины. По данным работы [10], формула (2) имеет строгое математическое доказательство в [12]. Она выражает теоретическую длину $\hat{N}$ программы через число входящих в ее код простых операторов η_1 и число простых операндов η_2 , исходя из минимаксного для данного типа программ так называемого стоимостного критерия. Это доказательство основано на предположении, что при создании программы реализуется функциональная задача в максимально экономной и рациональной форме. В [12] введено соотношение

$$N = \lambda (\eta_1 * \log_2 \eta_1 + \eta_2 * \log_2 \eta_2) = \lambda \hat{N}, \quad (3)$$

где λ – параметр, характеризующий степень приближения N к $\hat{N}$. Там же выполнено экспериментальное обоснование, в ходе которого – после обработки десяти профессионально составленных программ на языках высокого уровня – получено значение $\lambda = 0,934336314$, достаточно близкое к 1.

В литературе по методам программометрии используются и другие методы измерения характеристик программ. Тем не менее, программы, для ко-

торых значение λ приближается к 1, общепринято классифицировать как близкие по качеству к профессиональным программам, что и обосновывает выбор метрики М. Холстеда в данной работе.

Статистическая обработка программных текстов. После того, как все тексты АПП обработаны и для каждого из них уже вычислены соответствующие характеристики – экспериментальная и теоретическая длины, осуществляем обработку полученных параметров методами математической статистики.

Для выборки теоретических значений длин программ и экспериментально вычисленных значений длин программ, где объем каждой выборки (число обрабатываемых файлов) равен n , находим расчетные значения оценок статистических характеристик:

– математическое ожидание

$$\tilde{M}_x = \frac{1}{n} \sum_{i=1}^n x_i; \quad (4)$$

– дисперсия

$$\tilde{D}_x = \frac{\sum_{i=1}^n (x_i - \tilde{M}_x)^2}{n - 1}; \quad (5)$$

– среднее квадратичное отклонение

$$\tilde{\sigma}_x = \sqrt{\tilde{D}_x}. \quad (6)$$

Далее, когда оценка статистических характеристик выполнена, для теоретически полученной и экспериментально определенной длин АПП вычисляем коэффициент корреляции:

$$\tilde{r}_{xy} = \frac{\sum_{i=1}^n (x_i - \tilde{M}_x) \cdot (y_i - \tilde{M}_y)}{(n - 1) \cdot \tilde{\sigma}_x \cdot \tilde{\sigma}_y}. \quad (7)$$

Критерий совершенства Hq численно равен коэффициенту корреляции $\tilde{r}_{xy}$.

Для совершенных программ параметр Hq близок к 1, для программ, имеющих тот или иной набор несовершенств, отличия теоретического и экспериментально определенного значений длины текста будут более значительны, поэтому в результате коэффициент корреляции $\tilde{r}_{xy}$ уменьшится, то есть уменьшится параметр Hq .

Поскольку на начальном этапе создания программ особое внимание уделяется правильному, оптимальному построению алгоритма и реализации его на языке программирования, параметр Hq является важным для оценки качества АПП.

Программа оценки качественных характеристик АПП по методике Холстеда

Разработанная программа [13] представляет собой специализированную программу для автоматического определения качественных характеристик

академических программных продуктов на языке C++ на основе изложенной выше методики.

Программа реализована в среде Visual Studio на языке программирования C# и состоит из основной функции, функции лексического анализатора Analyzer() и функции синтаксического анализатора Result().

Процесс обработки программного текста организован следующим образом.

1. Основная функция программы считывает посимвольно текст каждого АПП из указанного пользователем подкаталога и отправляет каждый символ в лексический анализатор. Блок-схема алгоритма работы основной функции программы приведена на рисунке 1.

2. Лексический анализатор на основе поступающих на вход символов последовательно формирует лексему. Сформированная лексема передается в синтаксический анализатор.

3. Синтаксический анализатор сопоставляет входную лексему с грамматикой языка, которая описана в программном коде в виде статических строковых массивов, и определяет ее тип. В функции учитывается возможность ситуации, когда две различные переменные, находящиеся в разных областях видимости – локальная и глобальная, имеют одинаковое имя. Такие переменные фиксируются анализатором как два разных операнда. После того как определено, к какому классу относится поступившая на вход лексема, изменяется общее количество операторов или операндов программы, а также при необходимости пополняются словари операторов и операндов программы.

4. После полной обработки текста программы в основной функции программы вычисляется его экспериментальная и теоретическая длина.

После окончания обработки всех программных текстов АПП для всей группы текстов вычисляются оценки статистических характеристик (4)–(6).

Далее на основе полученных в (7) данных вычисляется критерий совершенства Hq . Результаты работы программы выводятся на экран.

Опишем взаимодействие программы и пользователя.

Для запуска процесса получения качественных характеристик пользователю необходимо выбрать подкаталог файлов – текстов АПП для их дальнейшей обработки. После того как пользователем указан необходимый подкаталог файлов, для каждого программного текста формируются и вычисляются следующие параметры качества: словарь операторов η_1 , словарь операндов η_2 , общее число всех операторов N_1 , общее число всех операндов N_2 , экспериментально полученная длина N и теоретическая вычисленная длина $\hat{N}$. Параметры выводятся на экран в виде таблицы. Под таблицей выводится вычисленное значение параметра совершенства для всей группы файлов. Наглядно это продемон-

стрировано на рисунке 2, где представлены результаты обработки АПП студентов специальности «Информационные системы» факультета математики и информационных технологий УлГУ, выполненных в 2013–2014 учебном году в рамках дисциплины «Технология программирования» по теме «Вычисления по формулам».

В таблице представлены экспериментальные данные, полученные с помощью разработанной программы при оценивании лабораторных работ по дисциплине «Технология программирования»

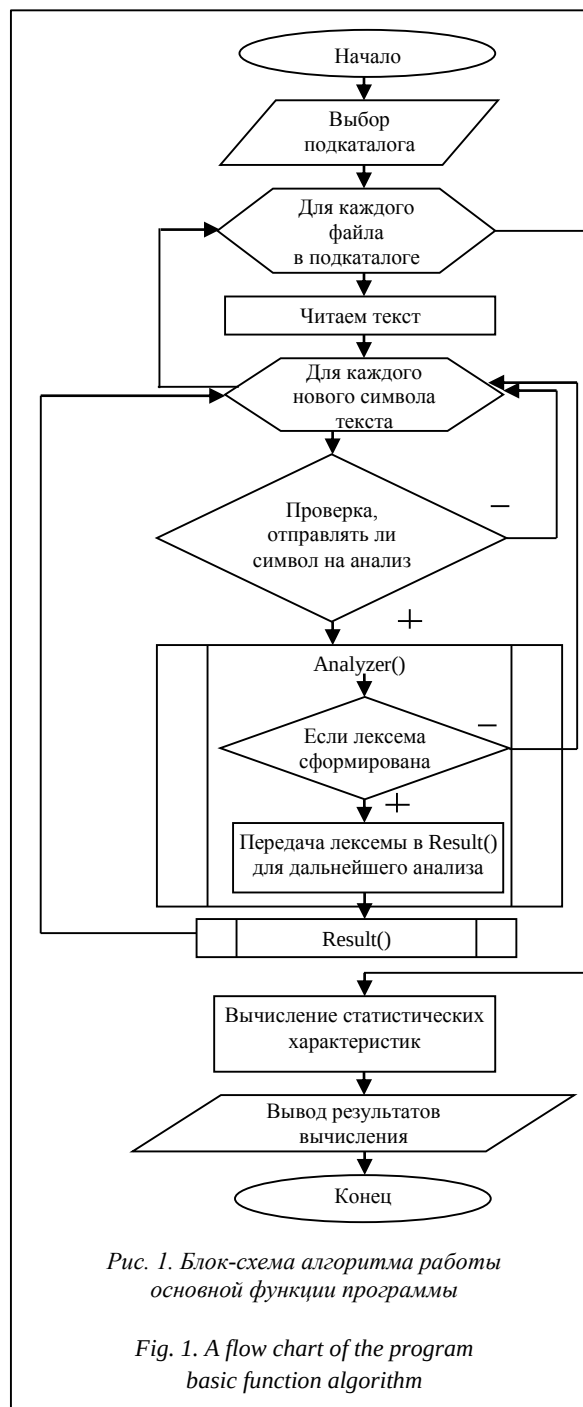
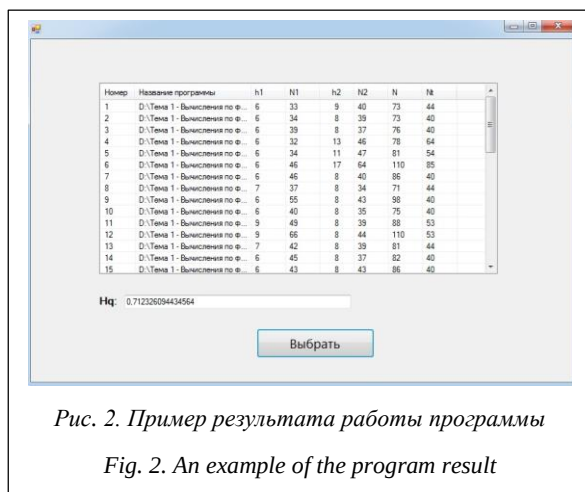


Рис. 1. Блок-схема алгоритма работы основной функции программы

Fig. 1. A flow chart of the program basic function algorithm



для студентов специальности «Информационные системы» факультета математики и информационных технологий УлГУ, выполненных также в 2013–2014 году. По количеству выполненных работ и параметру совершенства можно судить об уровне усвоения изученной темы. Например, в рамках тем 3–5 сдано мало работ, однако параметр качества для них высок. Из этого следует, что тему усвоило небольшое количество студентов, но, по-видимому, они отличники. Количество работ по теме 6 тоже невелико, а параметр совершенства значительно ниже, чем по всем предыдущим темам. Можно сделать вывод о сложности данной темы и необходимости усовершенствовать преподавание материала в рамках семинарских занятий.

Экспериментальные результаты определения параметра совершенства

Experimental results of perfection parameter determination

Тема	Количество работ	Параметр совершенства N_q
Вычисления по формулам	40	0,712326094434564
Операторы цикла	23	0,567967833381929
Одномерные массивы	8	0,851011658832649
Последовательности	3	0,96756114421808
Обработка текстовых строк	3	0,825354659308643
Матрицы	4	0,681555060203852

В заключение можно сделать следующие выводы. Объективные и проверяемые индикаторы качества АПП являются мощным инструментом модификации поведения студентов и преподавателей. Реализованные в специальных программах тестирования, они обладают рядом достоинств по сравнению с традиционными практиками: исключают риск манипулирования, обеспечивают многоаспектное понимание результатов деятельности кафедры, отдельного преподавателя и группы (потока) студентов, изучающих дисциплину, допускают прямое сравнение различных методик преподавания и являются публично доступными и легко интегрируемыми в учебный процесс. Применение таких индикаторов встречает положительные отзывы как со стороны преподавателей (снижает трудоемкость аттестации студентов, автоматизируя этот процесс), так и студентов (стимулирует стремление к достижению лучших показателей).

Созданная программа оценки качественных характеристик АПП на основе методики Холстеда [13] прошла успешную опытную эксплуатацию на материале лабораторных проектов одной из дисциплин «Технология программирования», в рамках которой было обработано 86 АПП, созданных в 2013–2014 учебном году.

Планируется продолжение опытной эксплуатации разработанной программы на расширенном наборе дисциплин, входящих в учебный процесс факультета математики и информационных технологий УлГУ, где учебным планом предусмотрены лабораторные работы в форме АПП. Для этого создается банк АПП. В настоящее время он содержит 151 запись.

Литература

1. Interface.ru. Interface and software company. URL: <http://www.interface.ru/home.asp?artId=3987> (дата обращения: 15.09.2015).
2. Антошина И.В., Домрачев В.Г., Ретинская И.В. Основные тенденции оценивания качества программных средств // Качество, инновации, образование. 2004. № 1. С. 70–75.
3. Калинина Л.Ю. Оценка качества программных продуктов // Качество, инновации, образование. 2006. № 4. С. 52–55.
4. Антошина И.В., Домрачев В.Г., Ретинская И.В. Методика составления системы характеристик качества для программных средств // Качество, инновации, образование. 2002. № 3. С. 57–60.
5. Липаев В.В. О проблемах оценивания качества программных средств // Качество, инновации, образование. 2002. № 1. С. 93–97.
6. Информационно-коммуникационные технологии в образовании. URL: http://www.ict.edu.ru/vconf/index.php?a=vconf&c=getForm&r=thesisDesc&d=light&id_sec=123&id_thesis=4643 (дата обращения: 15.09.2015).
7. Конгресс конференций «ИТО-2003». URL: <http://ito.edu.ru/2003/VI/VI-0-1548.html> (дата обращения: 15.09.2015).
8. Семушин И.В., Угаров В.В. Модификация поведения студента и преподавателя инженерных дисциплин // Московское научное обозрение. 2013. № 9 (37). С. 3–8.
9. Семушин И.В., Угаров В.В., Цыганова Ю.В., Афанасова А.И., Куличенко И.Н. Опыт проектно-ориентированного обучения в университетах Ульяновска // Перспективные информационные технологии (ПИТ 2014): тр. Междунар. науч.-технич. конф.; [под ред. С.А. Прохорова]. Самара: Изд-во Самар. науч. центра РАН, 2014. С. 436–438.
10. Угаров В.В. Компьютерные модели и программные комплексы в проектно-ориентированном обучении. Дисс. ... канд. техн. наук. Ульяновск, 2005.
11. Холстед М.Х. Начала науки о программах; [пер. с англ. В.М. Юфы]. М.: Финансы и статистика, 1981. 128 с.
12. Апостолова Н.А., Гольдштейн Б.С., Зайдман Р.А. О программнометрическом подходе к оценкам программного обеспечения // Программирование. 1995. № 4. С. 38–44.
13. Афанасова А.И. Программа оценки качественных характеристик академических программных продуктов на основе методики Холстеда: пат. 2015611297. Рос. Федерация, № 2014662403; заявл. 03.12.2014; зарег. 27.01.2015; опубл. 20.02.2015.

DOI: 10.15827/0236-235X.112.256-260

Received 16.09.15

HALSTEAD-TECHNIQUE BASED QUALITY ASSESSMENT FRAMEWORK FOR ACADEMIC SOFTWARE PRODUCTS

Afanasova A.I., Postgraduate Student, Junior Researcher, nastyaafanasova@yandex.ru
(Ulyanovsk State University, L. Tolstoy St. 42, Ulyanovsk, 432017, Russian Federation)

Abstract. Academic software products (ASP) assessment has a major impact on the acquisition level of complex computer and information disciplines. However, programmetry literature does not make the difference between ASP and professional software products (PSP). ASP projects, alternatively to PSP, do not have immediate achievement-oriented feedback from the market environment. Taking into account the scope of documentation, ASP frontal assessment comes out to be a labor-intensive work without necessary effect, which is a quick or visible accrue-ment in student competency on account of unavoidable instructor's subjectivity stemming from reliance on their personal experience, knowledge and beliefs. According to the authors, a solution is the creation of task-oriented methods, which are different from the PSP assessment methods, and in the development of specific instructor's tools for ASP assessment. Due to the created non-biased feedback, such tools can contribute to students' behavior modification. This will take the form of increased not only ASP quality, but also the quality of the entire project-based education process.

The paper presents the development results of such ASP assessment tool. Based on the Halstead technique, an algorithm to compute quantitative performance indices for the ASP created by students in C++ is set forth. The authors suggest an original computer program written in C# as a framework tool. The expected results are supported by the product beta testing and trial operation of the assessment framework.

Keywords: software quality assessment, student developed software, programmetry, Halstead metrics, project-based learning, C++.

References

1. *Interface.ru. Interface and software company.* Available at: <http://www.interface.ru/home.asp?artId=3987> (accessed September 15, 2015).
2. Antoshina I.V., Domrachev V.G., Retinskaya I.V. Basic software quality assessment trends. *Kachestvo, innovatsii, obrazovanie* [Quality, Innovations, Education]. 2004, no. 1, pp. 70–75 (in Russ.).
3. Kalinina L.Yu. Software quality assessment. *Kachestvo, innovatsii, obrazovanie* [Quality, Innovations, Education]. 2006, no. 4, pp. 52–55 (in Russ.).
4. Antoshina I.V., Domrachev V.G., Retinskaya I.V. Quality characteristics system technique for software means. *Kachestvo, innovatsii, obrazovanie* [Quality, Innovations, Education]. 2002, no. 3, pp. 57–60 (in Russ.).
5. Lipaev V.V. On software quality assessment problems. *Kachestvo, innovatsii, obrazovanie* [Quality, Innovations, Education]. 2002, no. 1, pp. 93–97 (in Russ.).
6. *Informatsionno-kommunikatsionnye tekhnologii v obrazovanii* [Information and Communication Technologies in Education]. Available at: http://www.ict.edu.ru/vconf/index.php?a=vconf&c=getForm&r=thesis-Desc&d=light&id_sec=123&id_thesis=4643 (accessed September 15, 2015).
7. *Kongress konf. "Informatsionnye tekhnologii v obrazovanii"* [Conf. Congress "Information Technologies in Education"]. Available at: <http://ito.edu.ru/2003/VI/VI-0-1548.html> (accessed September 15, 2015).
8. Semushin I.V., Ugarov V.V. Engineering student and teacher behavior modification. *Moskovskoe nauchnoe obozrenie* [Moscow Science Review]. 2013, no. 9 (37), pp. 3–8 (in Russ.).
9. Semushin I.V., Ugarov V.V., Tsyganova Yu.V., Afanasova A.I., Kulichenko I.N. Project-oriented education experience in Ulyanovsk universities. *Perspektivnye informatsionnye tekhnologii (PIT 2014): tr. Mezhdunar. nauch.-tekhnich. konf.* [Proc. of the Int. Science and Tech. Conf. "Prospect IT" (PIT 2014)]. S.A. Prokhorov (Ed.). Samara, Samara Scientific Center RAN Publ., 2014, pp. 436–438 (in Russ.).
10. Ugarov V.V. *Kompyuternye modeli i programmnye komplekсы v proektno-orientirovannom obuchenii* [Computer Models and Software Packages in Project-Oriented Education]. Ph.D. thesis. Ulyanovsk, 2005.
11. Halstead M.H. *Elements of Software Science*. NY, Elsevier North-Holland Publ., 1977 (Russ. ed.: Yufa V.M., Moscow, Finansy i statistika Publ., 1981, 128 p.).
12. Apostolova N.A., Goldshteyn B.S., Zaydman R.A. On a software metrics approach to software assessment. *Programmirovaniye* [Programming]. 1995, no. 4, pp. 38–44 (in Russ.).
13. Afanasova A.I. *Programma otsenki kachestvennykh kharakteristik akademicheskikh programmnykh produktov na osnove metodiki Kholsteda* [Academic Software Products Quality Assessment Program Based on Halstead Technique]. PC program. Patent RF no. 2015611297, Ulyanovsk State Univ., 2015 (in Russ.).

СОДЕРЖАНИЕ

Шукшунов В.Е., Янюшкин В.В.

Концептуальные основы разработки и создания учебно-тренажерно-моделирующего комплекса нового поколения 5

Ковригин С.Н., Янюшкин В.В.

Основы анализа IT-инфраструктуры открытого контура автоматизированной системы информационного обеспечения подготовки космонавтов для создания интегрированного тренажерного комплекса 16

Снытко А.С., Терников А.А.

Особенности модернизации центра обработки данных и космоцентра 22

Серов М.В., Кукин О.Н., Янюшкин В.В., Радченко В.М., Харагозян Р.К.

Стенд эргономической отработки пилотируемого транспортного корабля как средство проектирования и отработки интерфейсов экипажа 28

Радченко В.М., Бондарь Е.М., Чуланов А.О., Федоров Н.А., Лещина А.Е.

Применение систем виртуальной реальности при подготовке персонала к борьбе за живучесть 34

Кравченко С.И., Душенко А.Г., Андреев Д.А., Котов А.Г.

Активное информационное сопровождение действий экипажа 39

Кравченко С.И., Андреев Д.А., Брунько Д.В., Горбачев Е.Б.

Бортовой комплекс многофункциональной информационной системы Международной космической станции (Российский сегмент) 45

Кравченко С.И., Андреев Д.А., Долгов А.А., Колесников А.В.

Наземная подсистема подготовки электронной бортовой документации 49

Степанов В.В., Андреев Д.А., Геркен Е.А., Обыденков С.С.

Информационная поддержка космических экспериментов 55

Погорелов А.С., Андреев Д.А., Панфилов А.Н.

Сравнительный анализ и классификация задач по оптимальному расположению грузов в ограниченном пространстве 58

Бобков С.Г.

Проблемы создания высокотемпературных вычислительных систем 62

Мамросенко К.А., Решетников В.Н.

Моделирование подстилающей поверхности в имитационных системах 70

Михайлюк М.В., Торгашев М.А., Омельченко Д.В.

Эргономичный голосовой интерфейс управления антропоморфным роботом 75

Мадера А.Г.

Концепция математического и компьютерного моделирования тепловых процессов в электронных системах 79

Козлов Н.А., Бобков С.Г.

Высокопроизводительный блок интерфейса RapidIO для создания многоядерных микропроцессоров с виртуальными каналами RapidIO 87

Кулешов А.С.

Поддержка протокола MPI в ядре ОС Linux для многопроцессорных вычислительных комплексов на базе высокоскоростных каналов RapidIO 93

Тимохин П.Ю.

Система визуализации текстурированных моделей планет для тренировок проведения космических экспериментов 99

Трушин А.М.

Определение коллизий аппроксимирующих сфер и прямоугольных параллелепипедов в системах трехмерного моделирования 105

Власов А.О., Сурков А.В.

Маршрут проектирования самосинхронных конвейерных схем с использованием возможностей САПР 110

Болодурина И.П., Нугуманова А.А., Решетников В.Н.

Фильтр Калмана как метод вторичной обработки информации с системы ГЛОНАСС 116

Чекмарёв С.А. Инъектор сбоев для тестирования микропроцессоров типа система на кристалле к одиночным сбоям	121
Корнюшко В.Ф., Костров А.В., Породникова П.А. Подход к развитию системы управления тестированием программных средств	126
Есиков Д.О. Задачи обеспечения устойчивости функционирования распределенных информационных систем	133
Михайлов И.С., Зо Мин Тайк Реализация процесса поиска решения по модифицированному алгоритму Rete для нечетких экспертных систем	142
Чернышев Ю.О., Сергеев А.С., Рязанов А.Н., Капустин С.А. Разработка и исследование параллельного алгоритма муравьиных колоний для криптоанализа блочных криптосистем	148
Бутакова М.А., Гуда А.Н., Чернов А.В., Чубейко С.В. Оценка надежности программного обеспечения методами дискретно-событийного моделирования	158
Барabanов А.В., Марков А.С., Цирлов В.Л. Методический аппарат анализа и синтеза комплекса мер разработки безопасного программного обеспечения	166
Кораблин Ю.П., Шипов А.А. Эквивалентная характеристика формул LTL	175
Атаева О.М., Серебряков В.А. Основные понятия формальной модели семантических библиотек и формализация процессов интеграции в ней	180
Батура Т.В., Мурзин Ф.А., Проскуряков А.В. Программный комплекс для анализа данных из социальных сетей	188
Иванов Е.Ю., Косяков М.С. Репрезентативность метрик на основе событий процессора Intel Sandy Bridge при анализе времени обработки данных в памяти	198
Сорокин В.Е. Репликация данных в иерархических информационных системах с непостоянной связностью узлов	203
Дли М.И., Стоянова О.В., Белозерский А.Ю. Модели представления данных сложных производственных проектов в автоматизированных информационных системах промышленных предприятий	210
Грицай А.А. Интеллектуальная информационная система прогнозирования потребительского спроса и оптимизации товарных запасов	219
Иванова Е.И. Интеллектуальная информационная система для решения задач прогнозирования неисправностей вагонного оборудования на железнодорожном транспорте	231
Бурдо Г.Б., Федотова А.В. Алгоритмы и модели АСУ технологическими процессами технического обслуживания	237
Кохов Т.А., Корельштейн Л.Б., Гартман Т.Н. Программный комплекс проектирования обогрева технологических трубопроводов тепловыми спутниками для систем автоматизированного проектирования	244
Гусева Е.В., Сафаров Р.Р., Меньшуткина Н.В., Будран Ж. Подход к моделированию, масштабированию и оптимизации работы биореакторов на основании вычислительной гидродинамики	249
Афанасова А.И. Программа по оценке качества академических программных продуктов на основе методики Холстеда	256

C O N T E N T

Shukshunov V.E., Yanyushkin V.V. Conceptual framework for development of the next-generation educational training and simulation complex (etsc).....	5
Kovrigin S.N., Yanyushkin V.V. The analysis of an IT infrastructure of the astronaut training open automation system to create an integrated simulation complex	16
Snytko A.S., Ternikov A.A. Special features of data-processing center and space center upgrade	22
Serov M.V., Kukin O.N., Yanyushkin V.V., Radchenko V.M., Kharagozyan R.K. Ergonomic testing of a manned transport spacecraft for crew interface design and training	28
Radchenko V.M., Bondar E.M., Chulanov A.O., Fedorov N.A., Leshchina A.E. Virtual reality systems in damage control training of personnel	34
Kravchenko S.I., Dushenko A.G., Andreev D.A., Kotov A.G. Active information support of crew actions	39
Kravchenko S.I., Andreev D.A., Brunko D.V., Gorbachev E.B. The onboard multifunctional information complex of the International space station (Russian section)	45
Kravchenko S.I., Andreev D.A., Dolgov A.A., Kolesnikov A.V. The ground-based subsystem to prepare onboard electronic documentation	49
Stepanov V.V., Andreev D.A., Gerken E.A., Obydenov S.S. The information support for space experiments	55
Pogorelov A.S., Andreev D.A., Panfilov A.N. Comparison analysis and task classification according to cargo optimal placement in restricted space	58
Bobkov S.G. The problem of creating high temperature computing systems	62
Mamrosenko K.A., Reshetnikov V.N. Terrain modelling in simulation training systems	70
Mikhaylyuk M.V., Torgashev M.A., Omelchenko D.V. Ergonomic voice interface to control an anthropomorphic robot.....	75
Madera A.G. The concept of mathematical and computer simulation of thermal processes in electronic systems.....	79
Kozlov N.A., Bobkov S.G. High performance RapidIO block to create multi-core microprocessors with RapidIO virtual links.....	87
Kuleshov A.S. MPI support in Linux OS for RapidIO-based multiprocessor computer systems.....	93
Timokhin P.Yu. Textured planet model visualization system for space experiments simulation	99
Trushin A.M. Collision detection for bounding spheres and rectangular parallelepipeds in 3d modeling systems	105
Vlasov A.O., Surkov A.V. Automated design flow of self-timed pipelines using EDA tools	110
Bolodurina I.P., Nugumanova A.A., Reshetnikov V.N. Kalman filter as a method of secondary processing of GLONASS system information	116

Chekmaryov S.A. A fault injector to test the SoC-processor to single event upsets	121
Kornyushko V.F., Kostrov A.V., Porodnikova P.A. An approach to software testing management system development	126
Yesikov D.O. The problems of ensuring distributed information systems sustainability	133
Mikhailov I.S., Zaw Min Htike Finding sloutions by the modified Rete algorithm for fuzzy expert systems.....	142
Chernyshev Yu.O., Sergeev A.S., Ryazanov A.N., Kapustin S.A. Development and research of a parallel ant colony algorithm for a block cryptosystem cryptanalysis	148
Butakova M.A., Guda A.N., Chernov A.V., Chubeyko S.V. Software reliability evaluation by discrete-event simulation methods	158
Barabanov A.V., Markov A.S., Tsirlov V.L. A methodical framework of analysis and synthesis of secure software development controls	166
Korablin Yu.P., Shipov A.A. Equational characteristics of LTL formulas	175
Ataeva O.M., Serebryakov V.A. The basic concepts of a semantic libraries formal model and its integration process formalization.....	180
Batura T.V., Murzin F.A., Proskuryakov A.V. A software package for a social network data analysis	188
Ivanov E.Yu., Kosyakov M.S. Representativeness of Intel Sandy Bridge performance counters based metrics for memory access latency analysis	198
Sorokin V.E. Data replication in hierarchical information systems with unstable node connectivity	203
Dli M.I., Stoyanova O.V., Belozersky A.Yu. Complex industrial projects data representation models in automated information systems of industrial institutions.....	210
Gritsay A.A. Intelligent information system for retail demand forecasting and inventory optimization	219
Ivanova E.I. Intelligent information system for rolling stock carriage equipment faillures forecasting on railways	231
Burdo G.B., Fedotova A.V. Algorithms and models for maintenance process automated control systems	237
Kokhov T.A., Korelshteyn L.B., Gartman T.N. The software design package for technological pipelines heating by heat tracers in computer-aided design systems	244
Guseva E.V., Safarov R.R., Menshutina N.V., Boudrant J. The approach to modeling, scaling-up and optimization of bioreactors based on computational fluid dynamics	249
Afanasova A.I. Halstead-technique based quality assessment framework for academic software products	256