МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

ПРИБОРОСТРОЕНИЕ

ИЗДАНИЕ ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Журнал издается с января 1958 г.

ТОМ 51 **НОЯБРЬ 2008** **№ 11**

ТЕМАТИЧЕСКИЙ ВЫПУСК

КОМБИНИРОВАННЫЕ МЕТОДЫ И ТЕХНОЛОГИИ СИСТЕМНОГО МОДЕЛИРОВАНИЯ И АНАЛИЗА СЛОЖНЫХ ОБЪЕКТОВ

Под редакцией доктора технических наук, профессора Б. В. Соколова

СОДЕРЖАНИЕ

КОМПЛЕКСНОЕ МОДЕЛИРОВАНИЕ И АНАЛИЗ СЛОЖНЫХ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

THEMATIC ISSUE

COMBINED METHODS AND TECHNOLOGIES OF SYSTEMS MODELING AND ANALYSIS OF COMPLEX OBJECTS

By Edition of B. V. Sokolov, Doctor of Technical Science, Professor

CONTENTS

INTEGRATED MODELING AND ANALYSIS OF COMPLEX DYNAMIC OBJECTS

Editor-in-Chief L. F. Porfiriev

ПРЕДИСЛОВИЕ

Одна из важнейших особенностей развития науки во второй половине XX века и начале XXI века — возникновение ряда направлений, в которых центральными являются такие фундаментальные понятия, как система, информация, управление. В свете современных представлений эти понятия образуют концептуальное ядро новой отрасли научных знаний, которая получила в отечественной литературе название системной. Возникновение системной отрасли научных знаний является велением времени, так как на данном этапе развития науки (этапе интеграции научных знаний) на передний план выступает методология, требующая сочетания (единства) анализа и синтеза при изучении свойств сложных объектов и процессов как целостных образований, состоящих из взаимосвязанных частей и обладающих новыми свойствами по сравнению со свойствами этим частей. При этом в настоящее время речь должна идти не о взаимном поглощении, а о взаимном дополнении, концептуальном и идейном взаимообогащении, прежде всего, таких базовых междисциплинарных научных направлений, как общая теория систем, кибернетика и информатика.

Результаты исследований, подтверждающие плодотворность указанной междисциплинарной интеграции, непосредственно связаны с тематикой и содержанием статей, представленных в настоящем выпуске журнала. Авторы предлагаемых статей — сотрудники Санкт-Петербургского института информатики и автоматизации РАН (СПИИРАН), являющегося ведущим научным учреждением Отделения нанотехнологий и информационных технологий РАН в Северо-Западном регионе.

Статьи, представленные в данном номере журнала, сгруппированы в рамках четырех основных тематических направлений:

— интеллектуальные системы и технологии;
— модели и алгоритмы защиты информации;
— методы и средства обработки естественного языка;
— комплексное моделирование и анализ сложных динамических объектов.

Заместитель директора СПИИРАН по научной работе
доктор технических наук, профессор
Б. В. СОКОЛОВ

INTRODUCTION

Beginning of the whole series of directions, the main concepts of which are system, information, control, is one of the main peculiarities of science evolution during second part of the XX century — at the beginning of the XXI century. These concepts form conceptual focus which named as system branch. On the given science development phase (phase of the integrated scientific knowledge) methodology, requiring unity of the analysis and synthesis by study of complex-objects properties and processes as integral formations, consisting of interrelated parts and possessing new properties in comparison with properties of these parts, comes out in the foreground. At present conceptual addition such basic interdisciplinary scientific direction as general theory of systems, cybernetics and information science should be attended.

Investigations results connect directly to themes and the main contents of the papers collected in this issue of journal „Priborostroenie“. Authors of presented articles are members of the Saint-Petersburg Institute of Informatics and Automation (SPII) of Russian Academy of Science (RAS), which is leading scientific establishment of Department of Nanotechnologies and Information Technologies at the North-West region of Russia.

Offered articles are grouped within four basic thematic directions:

— intelligent systems and technologies;

— models and algorithms of information security;

— methods and means of natural-language processing;

— integrated modeling and analysis of complex dynamic objects.

Deputy Director of SPIIRAS,
Doctor of Technical Science, Professor
B. V. SOKOLOV

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

УДК 681.3

В. И. Городецкий, О. В. Карсаев, В. В. Самойлов, В. Г. Конюший

ЯЗЫК ОПИСАНИЯ МНОГОАГЕНТНЫХ СИСТЕМ

Приводится описание основ языка ASML, являющегося специфическим расширением „стандартного“ языка описания программного обеспечения UML и используемого в инструментальной среде MASDK 4.0 для проектирования многоагентных систем. Этап анализа и архитектурного проектирования систем выполняется с помощью графических диаграмм, предназначенных для описания метамоделей систем, протоколов взаимодействия, схем ролей и сценариев поведения ролей. Этап детального проектирования выполняется с помощью диаграмм, предназначенных для описания онтологий предметной области и сценариев поведения классов агентов.

Ключевые слова: *язык моделирования, многоагентные системы.*

Введение. Тенденции развития технологий разработки программного обеспечения определяются, с одной стороны, возможностями вычислительных средств, а с другой — потребностями индустриального сообщества. Интенсивное развитие Интернет- и Web-технологий постепенно приводит к качественному пересмотру концепции современного бизнеса и, как следствие, к усложнению задач, возлагаемых на программное обеспечение. В частности, одной из основных тенденций в настоящее время является интенсивное использование информационно-телекоммуникационной среды для разработки распределенных программных систем с открытой архитектурой. Такие системы состоят из множества сущностей (программных компонентов), каждая из которых решает свои локальные задачи и при необходимости взаимодействует с другими сущностями системы. Причем взаимодействия между сущностями системы становятся ключевым фактором, определяющим поведение и возможности всей системы в целом. Основное преимущество такой архитектуры — удобство моделирования с ее помощью реальных систем, что позволяет повысить скорость разработки программного обеспечения и расширить границы его практического применения.

Одним из наиболее перспективных подходов к разработке такого рода программных систем является многоагентный подход, при котором система моделируется множеством взаимодействующих интеллектуальных агентов. Возможности широкого применения многоагентного подхода в настоящее время ограничены относительно низким уровнем инструментальной поддержки процесса разработки многоагентных систем (МАС). Этим объясняется актуальность задачи создания инструментальной среды, поддерживающей полный цикл разработки МАС. При этом наиболее рациональным подходом к созданию такой среды является развитие базовых концепций объектно-ориентированного программирования и, в частности, создание специфических языков для описания моделей МАС на основе языка UML [1].

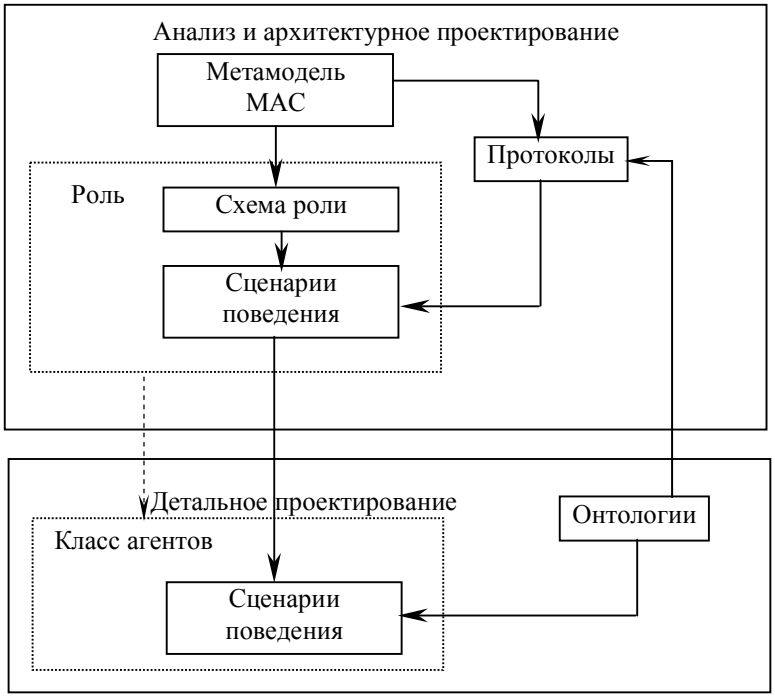
Наряду с развитием инструментальных сред проводятся исследования в области методологий разработки прикладных МАС. К числу наиболее широко известных и активно

развивающихся методологий относятся Gaia, MaSE, Tropos [1], среди которых Gaia [2] занимает особое место. Отличительная особенность этой методологии заключается в том, что в ней, по аналогии с организацией „живых систем“, основное внимание уделяется разработке и описанию организационных структур МАС.

Упомянутые два фактора, предполагающие использование методологии Gaia и специфического расширения языка UML для описания прикладных МАС, лежат в основе разработки инструментальной среды MASDK (Multi-Agent System Development Kit), развиваемой в СПИИРАН. При этом язык и методология имеют тесную взаимосвязь, обусловленную использованием в языке понятий и конструкций, определяемых в методологии Gaia.

Настоящая статья содержит изложение основ разработанного языка ASML (Agent-Based System Modelling Language), используемого для описания прикладных многоагентных систем четвертой версии среды MASDK. Описание предыдущей, третьей версии среды опубликовано в работе [3].

Визуальное проектирование МАС. Язык ASML по аналогии с языком UML имеет графическую нотацию, которая используется для визуального проектирования прикладных МАС. При этом разработка в среде MASDK 4.0 сводится к описанию моделей МАС с помощью диаграмм шести типов (см. рисунок). На этапе анализа и архитектурного проектирования используются диаграммы четырех типов, предназначенные для описания метамоделей МАС, протоколов взаимодействия, схем ролей агентов и сценариев поведения ролей агентов. Этап детального проектирования поддерживается диаграммами двух типов, предназначенных для описания онтологий предметных областей и сценариев поведения классов агентов.



Диаграммы и технология визуального проектирования

Проектирование МАС в среде MASDK 4.0 выполняется в соответствии с принципом „сверху вниз“. Это означает, что с помощью диаграмм первого типа описание метамодели МАС выполняется на системном уровне. Диаграммы последующих типов используются для детального описания элементов модели, введенных на первом или предыдущих уровнях.

Диаграммы метамоделей МАС. Для описания метамоделей МАС в языке ASML определены следующие классы понятий — *роли и классы агентов, активные сущности, цели (задачи), протоколы взаимодействия* и *сервисы*. При этом основополагающими элементами

метамоделей МАС служат понятия роли агентов и активные сущности. Описание ролей агентов является основой для идентификации классов агентов — эта задача относится к заключительной фазе описания диаграмм метамоделей МАС.

Содержательное описание *ролей агентов* выполняется с помощью классов понятий *метасценарии поведения* и *правила проактивного поведения*. Метасценарии поведения каждой роли, как правило, соответствуют задачам, которые выполняются данной ролью. Таким образом, в процессе идентификации метасценариев поведения, по сути, выявляется функциональность ролей. При этом агенты — как автономные сущности — могут инициировать выполнение своих метасценариев поведения самостоятельно, без каких-либо воздействий со стороны других агентов или пользователей системы. Для описания такого поведения ролей агентов используются классы понятий правила проактивного поведения. В текущей версии языка рассматриваются два типа правил такого рода: *временные* и *событийные*. Правила первого типа „срабатывают“ в зависимости от времени, правила второго типа — при наступлении соответствующих событий.

Класс понятий *активные сущности* используется для описания элементов системы, которые не являются агентами, а образуют их внешнюю среду: это могут быть, в частности, пользовательские приложения, различного рода сенсоры или активаторы.

Описываемые в метамодели МАС *цели (задачи)* могут относиться как к системе в целом, так и к отдельным ролям агентов.

Для описания метамоделей МАС используются следующие типы отношений: *решение задачи (достижение цели) инициируется активной сущностью / правилом проактивного поведения (роли); метасценарий поведения (роли) / активная сущность инициирует протокол взаимодействия; протокол взаимодействия / правило проактивного поведения инициирует метасценарий поведения (роли); протокол взаимодействия инициирует метасценарий поведения (роли) / активную сущность.*

С помощью отношений данных типов описывается схема взаимодействия ролей, которая предопределяется выбранной организационной структурой МАС.

Основным классом понятий, который позволяет описывать поведение МАС, подобное поведению „живых систем“, является *сервис*. Для этого класса понятий в языке ASML определены два взаимосвязанных типа отношений: *роль предоставляет / потребляет сервис, класс агента играет роль*. Описание сервисов и данных типов отношений обеспечивает организацию взаимодействия агентов в открытых системах. В частности, поиск агентов, предоставляющих необходимый сервис, лежит в основе того, как они формируют команды, необходимые для достижения целей системы.

Одним из результатов описания метамоделей МАС является идентификация элементов модели, которые должны быть детально описаны с помощью диаграмм других типов. Этими элементами являются *протоколы взаимодействия* и *метасценарии поведения (ролей).*

Диаграммы протоколов взаимодействия. Описание каждого протокола, идентифицированного при описании метамодели, выполняется с помощью отдельной диаграммы данного типа. Эти диаграммы являются расширением диаграмм последовательностей языка UML. При этом в языке ASML используются такие понятия, как *инициатор протокола, респондент протокола, коммуникационный акт, комбинированный фрагмент „Альтернативы“*. Последнее понятие применяется для описания фрагмента протокола, когда его участник в зависимости от текущей ситуации должен сделать выбор выполняемого коммуникационного акта из числа возможных, а также (в ряде случаев) выбрать, в адрес кого из участников протокола взаимодействия должен быть направлен тот или иной коммуникационный акт. Детальное описание самих коммуникационных актов выполняется с помощью языка ACL (Agent Communication Language) [4], принятого в качестве „стандарта“ для описания взаимодействий между агентами.

Диаграммы схем и сценариев поведения ролей. Диаграммы этих двух типов используются для детального описания метасценариев поведения ролей. Диаграммы схем ролей предназначаются для декомпозиции метасценариев поведения на несколько отдельных сценариев поведения и описания порядка их выполнения с помощью следующих типов отношений: *сценарий инициируется правилом проактивного поведения / протоколом взаимодействия; сценарий инициирует протокол взаимодействия; сценарий использует сценарий.*

Поведение агентов допускает параллельное выполнение двух и более метасценариев поведения. В связи с этим может возникнуть необходимость в координации выполнения разных сценариев поведения. Для этого в языке ASML используются класс понятий *событие* и связанные с ним следующие типы отношений: *сценарий генерирует событие; сценарий ожидает событие; правило проактивного поведения использует событие.* Первый тип отношений используется для идентификации сценария поведения, исполнение которого вызывает появление соответствующего события. Второй тип отношений используется для описания обработки прерываний выполнения сценариев, а именно: возникновение события является сигналом для продолжения сценария. Последний тип отношений описывает одно из условий срабатывания правил проактивного поведения — условие возникновения соответствующего события.

Диаграммы второго типа используются для детального описания сценариев поведения, идентифицированных при описании схем ролей. Это описание сводится к описанию *узлов сценариев* и их логических схем — переходов между узлами. В языке ASML определены классы понятий и отношений для описания следующих типов узлов сценариев: *действие, генерация / ожидание события; отправка / ожидание сообщений; обработка сообщения; сложное действие; узлы принятия решений* и др. Каждый из перечисленных типов узлов несет соответствующую функциональную нагрузку. В частности, в узлах принятия решений описываются возможные варианты выполнения сценариев поведения, в узлах типа сложное действие уточняется, когда выполняется вызов вложенных сценариев поведения, и т.д.

Диаграммы онтологий. Спецификация языка ASML также включает соответствующие классы понятий и типы связей между ними, необходимые для описания онтологий предметной области. Основными классами понятий являются *понятие предметной области* и *атрибуты понятий.* Описание атрибутов предполагает задание допустимых областей их значений. Описание онтологий позволяет устанавливать между понятиями следующие типы отношений: *наследование (обобщение), ассоциация* и *использование.*

Диаграммы сценариев поведения классов агентов. Диаграммы этого типа используются для детального описания поведения классов агентов. При этом базовая модель описания определяется: 1) множеством установленных при описании метамодели отношений типа *класс агента играет роль*; 2) описанием схем и сценариев ролей, связанных с классом агентов отношениями данного типа. Описание сценариев поведения класса агентов предполагает развитие сценариев поведения ролей, выполняемых данным классом агентов. Такое развитие сценариев не нарушает логических схем поведения, заданных для ролей. При этом рассматриваются две задачи.

Первая задача заключается в модификации описания сценариев поведения с учетом аспектов их реализации на уровне программного кода. Эта задача носит опциональный характер, т.е. необходимость в модификации сценариев поведения в ряде случаев может полностью отсутствовать, а если такая модификация и требуется, то она сводится к незначительным изменениям в описании сценариев поведения ролей.

Вторая задача состоит в детальной спецификации поведения класса агентов. Это предполагает описание *модельных переменных* и *параметров*, типы которых определяются в терминах понятий предметной области, описываемых в онтологиях. Для описания поведения на уровне классов агентов вводятся только модельные переменные, а на уровне сценариев — как модельные переменные, так и параметры. На основании этого формально определяются сиг-

натуры вызова сценариев поведения класса агентов. При этом сценарии могут вызываться как с уровня класса агентов, так и из других сценариев. В первом случае сценарии инициируются либо при срабатывании соответствующих правил проактивного поведения, либо в моменты начала участия агентов в протоколах взаимодействия в качестве респондентов.

Описание *модельных параметров* также выполняется и для детальной спецификации узлов сценариев поведения. Это позволяет формально специфицировать действия агентов, которые описываются в соответствующих узлах сценариев. Например, спецификация узлов принятия решений позволяет с помощью параметров формально описать условия возможных переходов.

Генерация программного кода. На основе созданного в процессе редактирования диаграмм описания модели МАС автоматически генерируется исходный программный код. При этом генерируются три библиотеки классов. Первая библиотека описывает поведение классов агентов на уровне сценариев поведения и их логических схем, вторая — структуру хранилища данных классов агентов и методы доступа к нему, третья — действия агентов, которые должны выполняться в каждом из узлов сценариев поведения. Первая и вторая библиотеки генерируются полностью, что исключает необходимость какого-либо их редактирования программистом. Третья библиотека содержит сгенерированное описание интерфейсов классов. При этом каждому сценарию поведения в этой библиотеке соответствует свой класс, а каждому узлу сценария — свой метод с описанием его параметров. Таким образом, для окончательного завершения разработки прикладной МАС на программном уровне необходимо реализовать методы третьей библиотеки.

Заключение. В настоящей статье описаны только основы языка ASML, используемого в инструментальной среде MASDK 4.0. Детальное описание языка, а также всей инструментальной среды в целом можно найти на Интернет-странице лаборатории интеллектуальных систем СПИИРАН [5].

Исследования, описанные в настоящей статье, осуществляются при поддержке Федерального агентства по науке и инновациям РФ (государственный контракт 02.514.11.4045).

СПИСОК ЛИТЕРАТУРЫ

1. *Bergenti F., Gleizes M. P., Zambonelli F.* Methodologies and Software Engineering for Agent Systems. Boston — Dordrecht — London: Kluwer Academic Publishers, 2004.

2. *Zambonelli F., Jennings N. R., Wooldridge M.* Developing multiagent systems: The Gaia Methodology // Transact. on Software Engineering and Methodology. 2003. Vol. 2, N 3. P. 317—370.

3. *Gorodetsky V., Karsaev O., Konushy V.* et al. Multi-agent system development kit // Software Agent-Based Applications, Platforms and Development Kits / Eds.: *R. Unland, M. Klusch, M. Calisti.* Birkhauser Book, 2005. P. 95—120.

4. Agent Communication Language Specification [Электронный ресурс]: <http://www.fipa.org/repository/aclspecs.html>.

5. Лаборатория интеллектуальных систем [Электронный ресурс]: <http://space.iias.spb.su/ai/index.jsp>.

Сведения об авторах

Владимир Иванович Городецкий — СПИИРАН, лаборатория интеллектуальных систем; E-mail: gor@iias.spb.su

Олег Владиславович Карсаев — СПИИРАН, лаборатория интеллектуальных систем; E-mail: ok@iias.spb.su

Владимир Владимирович Самойлов — СПИИРАН, лаборатория интеллектуальных систем; E-mail: samovl@iias.spb.su

Виктор Григорьевич Конюший — СПИИРАН, лаборатория интеллектуальных систем; E-mail: kvg@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 681.3

В. Г. КОНЮШИЙ, О. В. КАРСАЕВ

ИСПОЛЬЗОВАНИЕ АГЕНТСКОГО ПОДХОДА ПРИ КОНФИГУРИРОВАНИИ ВИРТУАЛЬНЫХ ПРЕДПРИЯТИЙ

Предлагается агентский подход к решению задачи календарного планирования работ при создании виртуальных предприятий в сфере нематериального производства. Подход заключается в представлении партнеров по виртуальному предприятию в виде множества независимых программных агентов. Приводятся результаты численных экспериментов, для которых использовались тестовые примеры из электронной библиотеки PSPLib.

Ключевые слова: *виртуальное предприятие, распределенное календарное планирование проектов, агентский подход.*

Введение. В последнее время для выполнения вполне определенных, зачастую уникальных проектов в сфере нематериального производства осуществляется создание виртуальных предприятий (ВП) ([1, 2]. Реализация проекта предполагает наличие общего координационного центра, роль которого выполняет руководитель проекта — один из партнеров ВП, выступающий, как правило, инициатором проекта. Другие партнеры являются исполнителями работ проекта и при этом могут быть задействованы в других проектах. Одна из основных задач, решаемых при создании ВП [3], — задача выбора партнеров и распределения среди них работ (заказов) т.е. задача конфигурирования ВП.

При решении задачи конфигурирования ВП могут быть использованы методы, применяемые при решении задач календарного планирования (КП) проектов. Суть задачи КП сводится к поиску а) варианта распределения работ среди потенциальных исполнителей и б) расписания выполнения работ, которое является оптимальным относительно заданных критериев. На этапе формирования ВП может быть рассмотрено несколько возможных планов выполнения проекта и/или несколько вариантов команд исполнителей работ — потенциальных партнеров ВП. В этом случае задача КП решается для каждого возможного варианта „план — команда“, и на основе сравнения полученных результатов выбирается наилучший вариант.

Большинство известных методов решения задач КП являются централизованными. Это предполагает, что вся необходимая информация о партнерах (их текущая загрузка, состояние ресурсов и т.п.) должна быть известна руководителю проекта, что, однако, не всегда возможно на практике. В связи с этим возникает задача реализации известных или разработка новых методов, позволяющих решать задачу КП в распределенном варианте. При этом использование интеллектуальных агентов для поддержки процессов создания и функционирования ВП является одним из наиболее перспективных подходов [4]. В данном случае ВП моделируется как сеть взаимодействующих независимых агентов, каждый из которых соответствует одному из партнеров ВП: руководитель проекта представлен агентом-ассистентом руководителя (далее по тексту — Руководитель), каждый исполнитель работ — агентом-ассистентом исполнителя (далее по тексту — Исполнитель).

В настоящей статье рассматривается использование агентского подхода для решения задачи КП, в которой для заданного плана работ и команды исполнителей необходимо найти вариант распределения работ среди Исполнителей и расписание их выполнения с учетом минимизации времени выполнения проекта.

Постановка задачи. План работ проекта описывается как множество работ $J = \{1, ..., n\} \cup \{0, n+1\}$, связанных отношением предшествования P. Работы 0-я и $(n+1)$-я обозначают начало и завершение всего проекта и являются фиктивными (не требуют времени и ре-

сурсов для выполнения). Отношение предшествования задается как множество пар $P = \{(i, j) \mid i \in J, j \in J\}$. Если $(i, j) \in P$, то работа j не может начаться раньше завершения работы i. Обозначим через P_j множество работ, предшествующих работе j, а через $E = \{1, ..., e\}$ — множество потенциальных Исполнителей. Для каждой работы $j \in (J \setminus \{0, n+1\})$ существует как минимум один потенциальный Исполнитель $e \in E$. Каждый Исполнитель e контролирует один тип возобновляемых ресурсов. Для выполнения каждой из работ требуется только один Исполнитель. Каждый Исполнитель может выполнять несколько работ одновременно.

Для j-й работы введем следующие обозначения: e_j — Исполнитель работы, $s_j \geq 0$ — момент начала работы, $f_j \geq s_j$ — момент ее окончания. Момент начала j-й работы равен максимальному времени завершения всех ей предшествующих P_j.

Расписанием (календарным планом) проекта будем называть совокупность троек $S = \{\langle s_0, s_0, \varnothing \rangle, \langle s_1, f_1, e_1 \rangle, ..., \langle s_n, f_n, e_n \rangle, \langle f_{n+1}, f_{n+1}, \varnothing \rangle\}$, заданных с учетом ограничений по ресурсам, которыми обладают Исполнители. В данной постановке задачи требуется найти календарный план S' с минимальным временем завершения проекта $T(S') = f_n$.

Изложенная постановка задачи КП известна в литературе как задача MRCPSP (Multi-Mode Resource-Constrained Project Scheduling Problem — задача календарного планирования проектов с ограниченными ресурсами и множеством режимов выполнения работ) [5]. Для этой задачи имеются тестовые наборы данных различной сложности, разработаны централизованные методы ее решения и известны наилучшие результаты применения этих методов. Далее в настоящей статье эти результаты используются для оценки эффективности предлагаемого децентрализованного подхода к решению данной задачи с использованием агентов.

Описание алгоритма. Предлагаемый алгоритм реализует эвристический подход к календарному планированию и основан на взаимодействии Руководителя проекта и потенциальных Исполнителей работ. Для организации взаимодействия использован известный протокол взаимодействия агентов „Сеть контрактов“ [6].

На каждом шаге алгоритма Руководитель определяет множество работ, готовых к назначению, анализирует предложения Исполнителей по выполнению этих работ и назначает Исполнителя для одной из них.

Задача Исполнителя $e \in E$ заключается в том, чтобы для каждой работы j, описание которой получено от Руководителя, сформировать предложение o_j^e по ее выполнению. Предложение включает данные о времени начала и окончания работы на основе оценки текущего состояния и ограничений использования своих ресурсов. Также предложения Исполнителей включают дополнительные данные — оценку величины простоя ресурсов.

Пусть A — множество работ, для выполнения которых Исполнитель уже известен и время задано; $R = J \setminus A$ — множество оставшихся работ; D — множество работ, готовых к назначению. Работа $j \in D$, если истинно условие $(j \in R) \wedge ((P_j = \varnothing) \vee (P_j \subseteq A))$.

Алгоритм поведения Руководителя проекта состоит в следующем.

1. Инициализация: $\forall j \in J : s_j = f_j = 0$.

2. Пока ($R \neq \varnothing$), выполнение следующих шагов:

   2.1. Найти множество допустимых работ D. Если $n+1 \in D$, то переход к шагу 3.

   2.2. Передать всем Исполнителям описание всех работ $j \in D$, в том числе, назначить самое раннее время начала работ $s_j = \max\{f_i \mid i \in P_j\}$.

   2.3. Получить от каждого Исполнителя множество предложений O_D^e по выполнению всех работ из множества D.

2.4. На основе анализа множества предложений $O_D = \{O_D^1 \cup ... \cup O_D^e\}$, полученного от всех Исполнителей, выбрать для назначения одну из работ $i \in D$.

2.5. На основе анализа множества предложений $O_i = \{O_i^1 \cup ... \cup O_i^e\}$ для работы i выбрать лучшее предложение $o_i^b \in O_i$ для работы i, где $b \in E$.

2.6. Назначить работу i Исполнителю b, приславшему предложение o_i^b.

2.7. Установить $s_i = s_i^b, f_i = f_i^b$, где s_i^b и f_i^b — время начала и окончания работы i, указанные в предложении o_i^b.

2.8. Обновить множество назначенных работ $A = A \cup i$.

3. Выход.

При выборе очередной работы для назначения и выборе Исполнителя этой работы Руководитель использует подход на основе правил приоритета [7, 8]. В ходе исследования рассматривались различные правила. В итоге были сформированы два комбинированных правила приоритета, состоящие из отдельных „простых“ правил выбора работы и ее Исполнителя.

Первое правило состоит в следующем: приоритет при выборе имеет работа, время начала которой минимально, а число оставшихся после нее работ до завершения проекта максимально. При назначении Исполнителя предпочтение отдается тому, кто предложил минимальное время окончания работы.

Второе правило отличается от первого только в части выбора Исполнителя: предпочтение имеет Исполнитель, который предложил минимальное время окончания работы и минимальное значение простоя ресурсов.

Правила предполагают рассмотрение нескольких критериев для выбора работы и Исполнителя. Абсолютные числовые значения, определяющие „веса“ отдельных критериев, вычисляются в соответствии с порядком применения простых правил, с помощью которых получены значения этих критериев. Найденные таким образом весовые коэффициенты отражают относительную важность или предпочтение на множестве критериев.

Значение приоритета работы (Исполнителя) вычислялось по формуле

$$Z = \frac{1}{\frac{1}{m}\sum_{k=1}^{m}\frac{k}{m}\mathrm{dif}_k}.$$

Здесь m — число критериев, а dif_k — нормализованное отклонение значения k-го критерия (V_k) от лучшего значения для этого критерия. Отклонение вычисляется по формуле

$$\mathrm{dif}_k = \frac{V_k - V_{\min}}{V_{\max} - V_{\min}},$$

где $V_{\min}$ и $V_{\max}$ — лучшее и худшее значения k-го критерия соответственно.

При вычислении значения приоритета рассматривался различный порядок применения простых правил. Если в итоге для нескольких работ или Исполнителей значения приоритетов совпадали, выбор производился случайным образом.

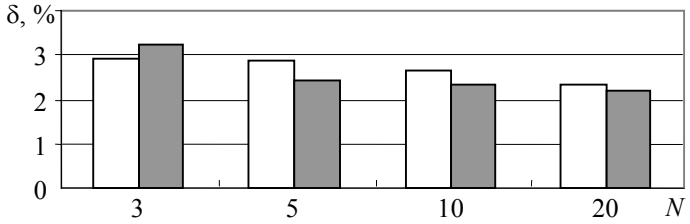
Описание экспериментов. Для тестирования описанного алгоритма были использованы тестовые примеры из набора N0 библиотеки PSPLib [9], которые предназначены для исследования методов решения задачи MRCPSP. При этом тестовые данные распределялись между агентами следующим образом. Руководитель получал только данные о структуре работ проекта, а каждому Исполнителю сообщались: 1) максимально доступный объем его ресурса; 2) данные о продолжительности каждой из работ; 3) требование к ресурсам в каждом из режимов выполнения (под режимом выполнения понимается определенное количество необходимых ресурсов).

Набор N0 состоит из 480 примеров. При проведении данного исследования использовались 240 примеров, в которых число типов ресурсов, необходимых для выполнения каждой работы, равно единице. Иными словами, для выполнения одной работы используются ресурсы только одного Исполнителя.

Основные характеристики тестовых примеров следующие. Количество работ (без учета начальной и конечной) — от 10 до 20. Число типов ресурсов — 2. Число режимов выполнения каждой из работ — 3. Коэффициент концентрации ресурсов $RS \in \{0{,}25,\ 0{,}50,\ 0{,}75, 1{,}0\}$*. Значение $RS = 0{,}25$ соответствует тестовым примерам с минимальным объемом имеющихся ресурсов, а $RS = 1{,}0$ — случаю с неограниченными ресурсами. Для всех примеров известно оптимальное решение (минимальная продолжительность проекта). С каждым из правил приоритета эксперименты проводились независимо, и для каждого из правил алгоритм повторялся многократно. В ходе экспериментов число повторных применений для одного правила составляло 3, 5, 10 и 20.

В результате для всех 120 примеров со значением коэффициента $RS = 0{,}75$ и $RS = 1{,}0$ были получены результаты, совпадающие с оптимальными, при 3- или 5-кратном повторении алгоритма. Для всех 60 примеров со значением коэффициента $RS = 0{,}5$, за исключением двух, оба правила позволяют получить результаты, совпадающие с оптимальными, при числе повторений алгоритма, равном 10.

Для наиболее сложных примеров ($RS = 0{,}25$) оптимальные результаты были получены только для половины из 60 примеров. Тем не менее результаты экспериментов, приведенные на рисунке, показывают, что предлагаемый распределенный подход позволяет получать решения, близкие к оптимальным, даже для очень сложных примеров. Так, средняя относительная погрешность (δ) полученных результатов, вычисленная относительно оптимальных значений, в зависимости от количества (N) повторений (3, 5, 10 или 20) алгоритма изменяется в интервале от 3 до 2 %. При этом максимальные значения относительной погрешности в отдельных примерах составили для первого правила 18 %, а для второго — 12 %.



Результаты экспериментов:
□ — правило 1, ■ — правило 2

Эксперименты проводились на персональном компьютере Pentium IV, 2800 МГц, 1024 Мб RAM. При 10-кратном применении алгоритма для определения результата при использовании первого правила среднее время вычисления составило 4,5 с, а второго — 8,7 с.

Заключение. Предложенный в настоящей статье подход к решению задачи календарного планирования с использованием агентов обеспечивает получение достаточно хороших результатов за ограниченное время. Причем в отличие от большинства известных методов данный подход позволяет решать задачи календарного планирования в распределенном варианте и поэтому может использоваться на практике, в частности, для организации виртуальных предприятий.

* Формальное определение коэффициента RS приведено в работе [9].

СПИСОК ЛИТЕРАТУРЫ

1. *Thoben K.-D., Eschenbächer J., Jagdev H. S.* Emerging concepts in E-business and extended products // E-Business Applications, Technologies for Tomorrow's Solutions / Eds.: *J. Gasos, K.-D. Thoben*. N.Y.: Springer-Verlag, 2003. P. 17—38.

2. *Гохберг Л.* Интеллектуальная деятельность — основа экономики информационного общества // Человек и труд. 2001. № 2. С. 32—34.

3. *Вютрих Х., Филипп А.* Виртуализация как возможный путь развития управления // Проблемы теории и практики управления. 1999. № 5. С. 94—100.

4. *Fischer K., Muller J.P., Heimig I., Scheer A.-W.* Intelligent agents in virtual enterprises // Proc. of the 1st Intern. Conf. and Exhibition on the Practical Applications of Intelligent Agents and Multi-Agent Technology, London, UK. 1996. P. 205—223.

5. *Kolisch R., Drexl A.* Local search for nonpreemptive multi-mode resource-constrained project scheduling // IIE Transact. 1997. Vol. 29, N 11. P. 987—999.

6. *Smith R. G.* The contract net protocol: High-level communication and control in a distributed problem solver // IEEE Transact. on Computers. 1980. Vol. 29, N 12. P. 1104—1113.

7. *Kolisch R.* Efficient priority rules for the resource-constrained project scheduling problem // J. of Operations Management. 1996. Vol. 14, N 3. P. 179—192.

8. *Kolisch R., Drexl A.* Adaptive search for solving hard project scheduling problems // Naval Research Logistics. 1996. Vol. 43, N 1. P. 23—40.

9. *Kolisch R., Sprecher A.* PSPLib — A project scheduling problem library // European J. of Operational Research. 1997. Vol. 96, N 1. P. 205—216.

Сведения об авторах

Виктор Григорьевич Конюший — СПИИРАН, лаборатория интеллектуальных систем; E-mail: kvg@iias.spb.su

Олег Владиславович Карсаев — СПИИРАН, лаборатория интеллектуальных систем; E-mail: ok@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 004.8

Н. Г. Шилов

ПОСТРОЕНИЕ КООПЕРАТИВНЫХ САМООРГАНИЗУЮЩИХСЯ СЕТЕЙ: ОСНОВНЫЕ ЗАДАЧИ И ТЕХНОЛОГИИ

Предложен подход, направленный на решение методологических проблем, возникающих при построении кооперативных самоорганизующихся сетей. Описаны задачи обеспечения взаимодействия между участниками самоорганизующихся сетей, а также изложены принципы выбора стандартов и протоколов для обмена информацией и ведения переговоров. Предложенный подход основан на концепции логистики знаний и включает в себя такие технологии, как управление онтологиями, профилирование и интеллектуальные агенты.

Ключевые слова: *самоорганизующиеся сети, многоагентная архитектура, Интернет-сервисы, протокол переговоров.*

Введение. Операции, связанные с ликвидацией последствий чрезвычайных ситуаций, как правило, подразумевают участие различных групп людей (зачастую многонациональных), которые должны сотрудничать в целях успешного достижения результатов. Такие группы могут включать в себя медицинские бригады, пожарные расчеты, спасательные команды, военные подразделения, представителей коммерческих / правительственных / некоммерческих организаций, добровольцев. Кроме того, в ходе этих операций может возникнуть необходимость использования внешних информационных источников (например, медицинских баз данных, служб, предоставляющих информацию о доступности средств транспортировки, прогнозе погоды и т.п.). Эффективность работы организаций, сформированных на основе коалиций, обусловлена такими требованиями, как интенсивный обмен информацией для достижения необходимого уровня „понимания“ ситуации (situation awareness), создание планов действий „на ходу“, а также обеспечение участников актуальной информацией.

Централизованный подход не всегда применим, например, вследствие возможных повреждений инфраструктуры или различной субординации участников. Серьезным недостатком централизованного управления может стать отказ центрального управляющего узла, что повлечет за собой остановку всей операции. Возможным решением данной проблемы является организация децентрализованных кооперативных самоорганизующихся коалиций, состоящих из участников операции [1]. Однако для работы такой сети необходимо решить ряд проблем, которые можно условно разделить на технические и методологические.

В настоящей статье рассматривается подход, направленный на решение методологических проблем, возникающих при построении кооперативных самоорганизующихся сетей [2, 3]. Главным образом, описаны задачи обеспечения взаимодействия между их участниками, а также приведены принципы выбора стандартов и протоколов для обмена информацией и ведения переговоров. Предложенный подход основан на разработанной ранее концепции логистики знаний [4] и включает в себя такие технологии, как управление онтологиями, профилирование и интеллектуальные агенты.

Основные принципы построения самоорганизующихся сетей. На первом этапе исследования были определены и проанализированы фазы жизненного цикла самоорганизующихся сетей, основные задачи, требующие решения, а также сервисы и информационные модели (табл. 1).

Таблица 1

Фаза жизненного цикла	Задачи	Сервисы и модели
Построение сообщества (единожды, новые участники могут появляться в любое время)	Формирование общей инфраструктуры Определение общих стандартов и протоколов взаимодействия/ведения переговоров	Идентификация, оценка, регистрация участников Общая модель представления знаний Общая модель участников сообщества
Формирование сети (непрерывное, инициируется изменениями в текущей ситуации)	Описание задач Выбор партнеров	Модель описания задач (контекст) Правила принятия решений при выборе партнеров
Функционирование (непрерывное)	Координация и синхронизация	Правила изменения решений и проведения повторных переговоров
Прекращение работы	Завершение установленных соглашений	Обновление текущего решения

На основе данного анализа были сформулированы следующие основные принципы рассматриваемого подхода.

1. Общая совместно используемая онтология верхнего уровня служит для унификации терминологии. Каждый участник работает со своим фрагментом этой онтологии, соответствующим его возможностям / задачам. Данные фрагменты синхронизируются автоматически при необходимости. Определение правил формирования фрагментов является задачей предстоящих исследований. Возможны два направления: 1) построение общей подробной онтологии, которая будет использоваться всеми участниками сети; 2) динамическое обновление знаний участников при поиске партнеров и ведении переговоров (общая подробная онтология отсутствует). Предлагаемый в статье подход представляет собой комбинацию обоих направлений.

2. Каждому участнику приписывается профиль, описывающий его способности.

3. Каждому участнику назначается представляющий его интеллектуальный агент, который собирает информацию, необходимую для „понимания“ ситуации участником, и ведет переговоры с другими агентами для создания совместных планов действий в зависимости от текущей ситуации. При ведении переговоров агент должен следовать определенным правилам, зависящим от роли конкретного участника.

4. Для взаимодействия участников и обмена информацией должны использоваться стандарты Интернет-сервисов, такие как OWL (Web Ontology Language). Внешние источники также должны поддерживать эти стандарты и терминологию, определенную в общей онтологии.

Для оценки возможности применения предлагаемого подхода планируется использовать сценарий ликвидации последствий чрезвычайной ситуации и эвакуации пострадавших. Сценарий включает в себя такие задачи, как конфигурирование и размещение портативных госпиталей, доставка комплектующих к местам размещения госпиталей, эвакуация людей с места катастрофы. Подробное описание сценария приведено в работе [5]. Оценку подхода планируется проводить посредством разработки научно-исследовательского прототипа информационной системы, реализующей данный подход, и выполнения экспериментов на основе разработанного сценария.

Многоагентная архитектура самоорганизующихся сетей. Общая схема самоорганизующейся сети приведена на рис. 1. Каждый участник операции представлен в системе назначенным ему агентом. Архитектура агента представлена на рис. 2. Каждый агент обладает знаниями, хранящимися в его базе знаний. Эти знания описываются фрагментом общей онтоло-

гии, относящимся к задачам и способностям данного участника (и соответственно его агента), и называются контекстом участника. Способности, предпочтения и другая информация об участнике хранятся в его профиле, доступном для просмотра агентам других участников сети. Профиль позволяет ускорить взаимодействие, которое осуществляется посредством коммуникационного модуля, отвечающего за выполнение требований, предъявляемых протоколами и стандартами, используемыми в сети.

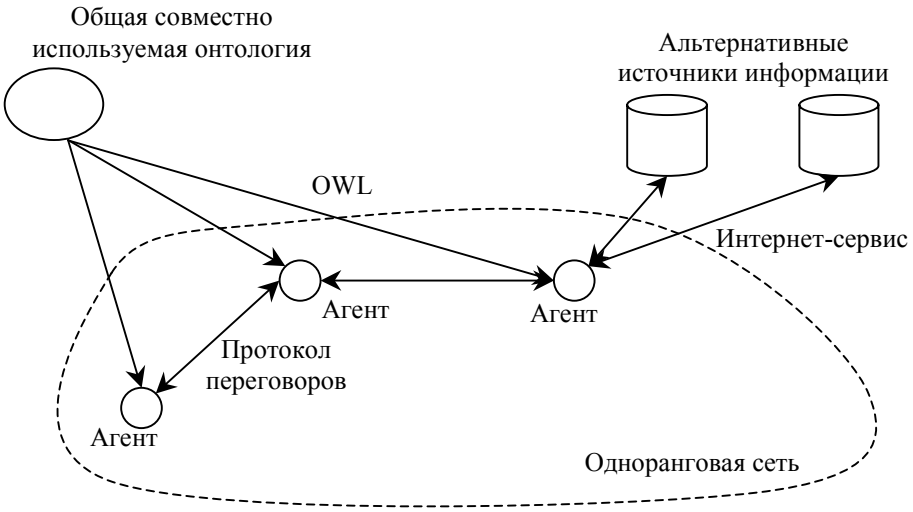


Рис. 1. Общая схема самоорганизующейся сети

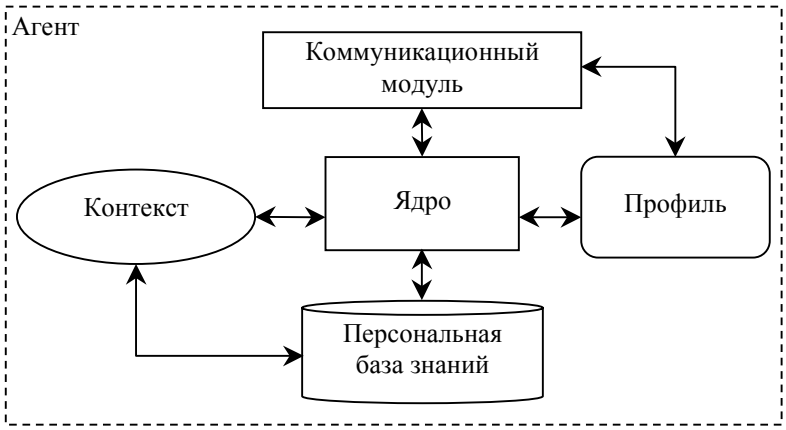


Рис. 2. Архитектура агента

Взаимодействие агентов осуществляется с двумя основными целями: 1) установление связей и обмен информацией для лучшего „понимания“ ситуации, 2) ведение переговоров и заключение соглашений для координации действий во время операций. Агенты могут также получать информацию из различных источников, например, информация о местной сети дорог может быть получена из геоинформационной системы.

Протокол переговоров. Быстрое принятие правильных решений в операциях по ликвидации чрезвычайных ситуаций и эвакуации пострадавших является очень важным. Поскольку временные ограничения не всегда позволяют выполнить оптимизацию для поиска наилучших решений, имеет смысл говорить о допустимых решениях. Ниже приведены основанные на предлагаемом подходе требования, которые необходимо учитывать при выборе протокола переговоров.

1. *Вклад*: агенты должны координировать свои действия в целях достижения максимальной эффективности работы всей сети, а не извлечения собственной выгоды.

2. *Выполнение задачи*: основной целью является выполнение поставленной задачи, а не извлечение выгоды.

3. *Одноранговые переговоры*: агенты функционируют в децентрализованном сообществе, и большинство переговоров проводятся в отсутствие агентов-лидеров, управляющих переговорным процессом и выносящих окончательное решение.

4. *Общая терминология*: поскольку агенты работают в рамках единой системы, они используют единую терминологию, определяемую общей онтологией.

5. *Доверие*: поскольку агенты работают в рамках единой кооперативной сети, они могут доверять друг другу (т.е. нет необходимости проверять информацию, получаемую от других агентов).

Для выбора базового протокола переговоров агентов были проанализированы классические протоколы (табл. 2, здесь „v“ и „–“ обозначают соответственно удовлетворение и невозможность удовлетворения требований). На основе анализа этих протоколов и вышеуказанных требований к ним в качестве базы для разработки протокола переговоров были выбраны „торги“ и „сети контракторов“.

Таблица 2

Требование	Протокол					
	Голосование	Торги	Аукционы	Механизмы рыночного равновесия	Коалиционные игры	Сети контракторов
Вклад	v	v	–	–	– / v	v
Выполнение задачи	– / v	– / v	–	–	–	v
Одноранговые переговоры	–	– / v	v	–	–	–
Общая терминология	v	v	v	v	v	v
Доверие	v	v	–	v	v	v

Заключение. Изложенный в статье подход к построению самоорганизующихся сетей позволяет избежать ряда недостатков централизованного управления.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант № 06-07-89242.

СПИСОК ЛИТЕРАТУРЫ

1. *Viana A. C., Amorim M. D., Fdida S., Rezende J. F.* Self-organization in spontaneous networks: the approach of DHT-based routing protocols // Ad Hoc Networks J., Special Issue on Data Communications and Topology Control in Ad Hoc Networks. 2005. Vol. 3, N. 5. P. 589—606.

2. *Hammer B., Micheli A., Sperduti A., Strickert M.* Recursive self-organizing network models // Neural Networks. 2004. Vol. 17, N. 8—9. P. 1061—85.

3. *Nakano T., Suda T.* Self-organizing network services with evolutionary adaptation // IEEE Transact. on Neural Networks. 2005. Vol. 16, N. 5. P. 1269—1278.

4. *Smirnov A., Pashkin M., Levashova T., Chilov N.* Fusion-based knowledge logistics for intelligent decision support in network-centric environment // Intern. J. of General Systems. 2005. Vol. 34, N 6. P. 673—690.

5. *Smirnov A., Levashova T., Shilov N.* Context-driven decision support for megadisaster relief // J. of Emergency Management. 2006. Vol. 4—5. P. 51—56.

Сведения об авторе

Николай Германович Шилов — СПИИРАН, лаборатория интегрированных систем автоматизации; E-mail: nick@iias.spb.su

Поступила в редакцию
06.05.08 г.

МОДЕЛИ И АЛГОРИТМЫ ЗАЩИТЫ ИНФОРМАЦИИ

УДК 004.056

В. В. Воронцов, И. В. Котенко

АНАЛИЗ МЕХАНИЗМА ОБНАРУЖЕНИЯ И СДЕРЖИВАНИЯ ЭПИДЕМИЙ СЕТЕВЫХ ЧЕРВЕЙ НА ОСНОВЕ „КРЕДИТОВ ДОВЕРИЯ“

Обсуждаются вопросы, связанные с анализом и модификацией механизма обнаружения и сдерживания эпидемий сетевых червей, который базируется на так называемых „кредитах доверия“. Представлены особенности реализации данного механизма защиты, а также методика и результаты его оценки для различных типов сетевого трафика.

Ключевые слова: *сетевая безопасность, сетевые черви, обнаружение и ограничение распространения сетевых червей, механизмы защиты, основанные на кредитах доверия, моделирование защиты от сетевых червей.*

Введение. В настоящее время рост числа инцидентов, связанных с применением различных злонамеренных программ, продолжает увеличиваться. Очевидно, что одну из основных угроз в сложившейся ситуации, наряду с компьютерными вирусами, представляют сетевые черви. Это объясняется не только возросшим уровнем компьютерной грамотности злоумышленников, в том числе и в области компьютерной безопасности, но и эффективностью применения таких программ злоумышленниками для реализации преследуемых ими целей.

С учетом этого актуальность задачи разработки механизмов обнаружения и ограничения распространения сетевых червей не вызывает сомнений. Одним из методов обнаружения сетевых червей являются механизмы, которые используют различные схемы проверки статистических гипотез. Среди них можно выделить механизмы, основанные на „пороговом случайном прохождении“ (Threshold Random Walk — TRW) [1] и кредитах доверия (Credit Based Rate Limiting — CB) [2]. Данный класс механизмов предназначен для обнаружения быстро распространяющихся сетевых червей. Однако область его применения и его эффективность исследованы в недостаточной степени, поэтому анализ данного механизма и его адаптация к различным видам сетевого трафика и конфигурациям компьютерных сетей является важной и актуальной научной задачей в области компьютерной безопасности.

В настоящей статье приводятся описание предлагаемой методики моделирования и оценки механизма обнаружения и ограничения распространения сетевых червей на основе кредитов доверия, а также результаты моделирования работы этого механизма.

Механизм обнаружения и сдерживания на основе кредитов доверия. Класс механизмов, основанных на кредитах доверия [2], для обнаружения факта инфицирования узла использует идею анализа статистических данных о неудачных соединениях, предложенную в работе [1]. Основная модификация механизма, рассмотренного в работе [1], заключается в изменении направления анализа поступивших событий на хронологически обратный [2], т.е.

наступившие события анализируются в обратном порядке — наиболее поздние анализируются в первую очередь, за счет чего сокращается время обнаружения факта инфицирования узла.

При функционировании механизма защиты, основанного на кредитах доверия, для каждого узла сети составляется список (PCH-лист, PCH — Previous Contacted Hosts) ранее посещенных узлов (т.е. узлов, к которым были осуществлены первоначальные обращения с данного узла). Запись о первоначальном соединении содержит следующие поля: *IP*-адрес узла, попытку установления соединения с которым необходимо осуществить, время попытки установления соединения, состояние записи („ожидание“, „успешно“ или „ошибка“). Каждому узлу также ставится в соответствие очередь первоначальных соединений (FCC-очередь, FCC — First Contact Connection), в которой хранятся все поступившие запросы от этого узла. Запросы из этой очереди обрабатываются в порядке их поступления, вне зависимости от статуса.

При получении системой мониторинга запроса на соединение от узла I проверяется, существует ли *IP*-адрес узла, с которым осуществляется попытка установления соединения, в списке ранее посещенных узлов узла I. При отсутствии адреса назначения в этом списке он добавляется в PCH-лист узла I, и в очередь FCC добавляется соответствующая запись с адресом узла назначения в качестве параметра и статусом „ожидание“.

При поступлении запроса, обращенного к узлу I, от узла, запрос к которому находится в очереди FCC узла I, данный запрос считается ответом на запрос первоначального соединения, и статус записи устанавливается в состояние „успешно“, за исключением случая, когда ответом является пакет TCP RST.

При нахождении записи в начале очереди со статусом „ожидание“ в течение временного интервала, превышающего тайм-аут, предусмотренный протоколом при запросе соединения, по истечении этого периода записи присваивается статус „ошибка“.

Далее исходящий трафик анализируется на предмет возможности заражения червем при помощи метода обратной последовательной проверки гипотез [1].

Работа механизма сдерживания на основе кредитов доверия базируется на следующих двух принципах: 1) исходящие запросы от узла разрешаются только при положительном значении переменной, задающей количество кредитов, в противном случае запросы на соединение блокируются; 2) производится периодическая корректировка количества доступных кредитов.

Необходимость периодической корректировки количества доступных кредитов обусловлена возможностью возникновения опасной ситуации: большое количество кредитов может вызвать задержку в обнаружении червя, или неполадки сети могут привести к тому, что неинфицированный узел истратит все свои кредиты и тем самым станет заблокированным.

Для разрешения этой ситуации используются две эвристики. Во-первых, если узел остается без кредитов больше заданного времени, ему присваивается один кредит. Предоставление кредита происходит гарантированно после принятия решения о том, что узел не инфицирован. В случае признания узла инфицированным, он будет заблокирован (таким образом, дальнейшее распространение червя будет прекращено независимо от количества кредитов). Во-вторых, каждую секунду работы механизма узел, имеющий больше заданного количества кредитов, обязан отдать треть от их числа, но с условием, что оставшееся количество кредитов не меньше изначально установленного. Как показано в работе [2], узел, демонстрирующий большое количество успешных соединений, в любой момент имеет количество кредитов, в два раза превышающее частоту установления соединений для этого узла.

Методика моделирования и оценки механизма защиты. Для оценки эффективности исследуемого механизма и его усовершенствования был разработан и реализован программный комплекс моделирования работы механизмов обнаружения и ограничения распространения сетевых червей [3, 4].

Разработанный комплекс позволяет моделировать реакцию механизма защиты на сетевой трафик как при наличии червя в компьютерной сети, так и при его отсутствии. Функционирование исследуемого механизма защиты оценивается по следующим показателям эффективности методов обнаружения и реагирования: доля блокированного и (или) задержанного легитимного трафика (степень ложных срабатываний, false positives); доля пропущенного злонамеренного трафика (степень пропусков атак, false negatives); время реакции на атаку и др.

Обобщенная методика моделирования и оценки исследуемого механизма защиты представлена на рис. 1.

Рис. 1. Обобщенная методика моделирования и оценки механизма защиты

В начале процесса моделирования выбирался тип „чистого“ сетевого трафика (при отсутствии червей) для тестирования и последующей оценки эффективности работы исследуемого механизма. В качестве базовых использовались записи сетевых трафиков следующих типов: нормального сетевого трафика (без „быстрых“ приложений и червей); сетевого трафика при условии функционирования в сети приложений P2P (peer-to-peer) как примера быстрых приложений; комбинация этих двух типов. Записи трафиков были получены как из открытых источников, так и из созданной тестовой сети.

Задача получения трафика червя для дальнейшего комбинирования с „чистым“ трафиком была решена посредством генерирования сетевых пакетов с параметрами, соответствующими той или иной модели червя. При моделировании использовались модели известных сетевых червей (например, Code Red II, Slammer) и модели потенциально возможных („искусственных“) червей, формируемые на основе задания таких параметров функционирования, как тип соединения (TCP или UDP); частота генерации пакетов; изменение скорости, с которой производится сканирование; тип сканирования или методика выбора адреса узла-получателя и порта; вероятность установления успешного TCP-соединения; размер пакета и др.

Затем устанавливались параметры исследуемого механизма обнаружения, и для комбинаций различных трафиков проводилось моделирование работы тестируемого механизма.

В результате моделирования фиксировались значения наиболее важных показателей эффективности работы исследуемого механизма: процентное соотношение ложных срабатываний, доля пропущенного трафика червя, время реакции на атаку, интенсивность работы с памятью,

а также максимальный объем памяти, необходимый при работе тестируемого механизма. Далее оценивалась эффективность работы механизма при заданных исходных параметрах и, в случае необходимости, проводился анализ степени влияния на эффективность механизма защиты каждого из исходных параметров (с дальнейшей корректировкой их значений).

Результаты экспериментов и предложения по использованию механизма защиты. Было проведено более 700 экспериментов для различных видов трафика и разных исходных параметров.

На рис. 2 показана зависимость суммарной ошибки (Σ) работы механизма обнаружения и сдерживания на основе кредитов доверия от процента успешных соединений в трафике (N) при отсутствии в сети сетевого червя (на „чистом“ трафике). Как видно из рисунка, исследуемый механизм продемонстрировал в целом приемлемое качество работы — было заблокировано в среднем менее 5 % нормального трафика. На рис. 3. проиллюстрирована зависимость $\Sigma(N)$ при наличии в анализируемом трафике трафика сетевого червя.

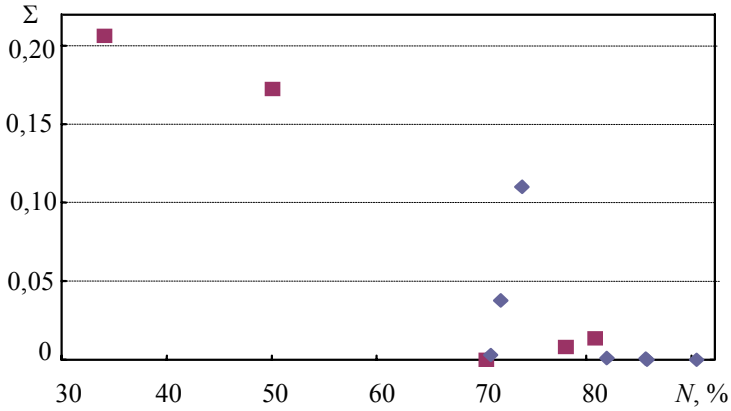


Рис. 2. Результаты экспериментов для „чистого“ трафика:
■ — P2P-трафик, ◇ — нормальный трафик

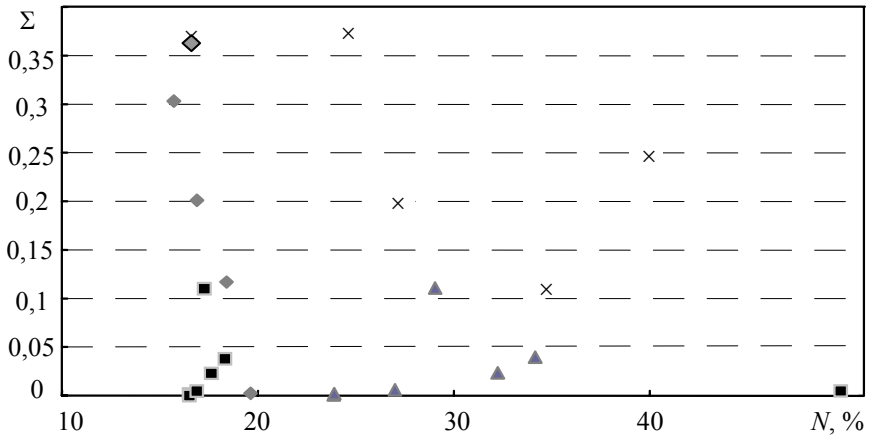


Рис. 3. Результаты экспериментов для смешанного трафика:
◇ — комбинация трафика искусственного червя и P2P-трафика,
■ — комбинация трафика искусственного червя и нормального трафика,
△ — комбинация трафика червя Code Red II и нормального трафика,
× — комбинация трафика червя Code Red II и P2P-трафика

Результаты моделирования можно оценить в двух случаях: при наличии или отсутствии в сети быстрых приложений. Так, при исследовании комбинации „нормальный трафик — трафик червя“ при отсутствии в сети активной работы „быстрых“ приложений (в рассматриваемом случае P2P-клиентов) результаты тестирования продемонстрировали положительный эффект применения механизма обнаружения и сдерживания сетевых червей. Несмотря на то что в наихудшем случае суммарная ошибка превысила значение 0,1 (т.е. было заблокировано

около 10 % трафика), процент пропуска атаки не превысил 0,17 для искусственного червя и составил около 1 при наличии в сети червя Code Red II.

Не столь оптимистичная картина была получена по результатам тестирования механизма для комбинации „быстрый трафик — трафик червя“, т.е. при активном функционировании в сети P2P-клиентов. Максимальная суммарная ошибка в этом случае составила от 0,37 до 0,385 для трафиков искусственного червя и червя Code Red II соответственно. Такое большое значение ошибки было вызвано, прежде всего, большой степенью ложного срабатывания — 0,3635 и 0,384 соответственно.

Таким образом, очевидно, что основная проблема при использовании исследуемого метода заключается в блокировании большого процента легального трафика. Объяснением этому служит то, что поведенческая модель „быстрых“ приложений идентична поведенческой модели сетевого червя.

Исходя из этого дальнейшее усовершенствование предлагаемого метода целесообразно вести в направлении улучшения качества дифференциации последовательностей пакетов „быстрых“ приложений и пакетов сетевого червя.

Представляется, что процент блокировки легального трафика может быть уменьшен за счет анализа дополнительных признаков сетевых пакетов. Например, могут быть проанализированы такие признаки, как размер пакета, некоторые скоростные характеристики обмена, порты, по которым устанавливаются соединения, и т.д.

Выбор этих параметров обусловлен следующими соображениями:

1) для сеансов „быстрых“ приложений, вероятнее всего, характерны определенные шаблоны действий, например, посылка большого количества пакетов одинакового размера на одни и те же порты с той или иной периодичностью;

2) при анализе отдельных пакетов необходим дополнительный анализ данных, содержащихся в пакете; это потребует увеличения расхода системных ресурсов для работы алгоритма, что приведет к невозможности аппаратной реализации исследуемого механизма.

Заключение. Итак, в настоящей статье проанализирован один из перспективных механизмов защиты от сетевых червей — механизм обнаружения и сдерживания сетевых червей на основе кредитов доверия, описана общая методика моделирования и оценки механизмов защиты, приведены результаты экспериментов. Проанализированы положительные и отрицательные стороны существующего механизма [2] и предложены направления для дальнейшей его модификации в целях повышения качества обнаружения и сдерживания сетевых червей при условии, что в защищаемой подсети активно функционируют так называемые „быстрые“ приложения.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 07-01-00547), программы фундаментальных исследований ОИТВС* РАН (контракт № 3.2/03), Фонда содействия отечественной науке, а также при частичной финансовой поддержке, осуществляемой в рамках проекта с фирмой „Hewlett-Packard“ (США) и проекта Евросоюза RE-TRUST (контракт № 021186-2).

СПИСОК ЛИТЕРАТУРЫ

1. *Jung J., Paxson V., Berger A.W., Balakrishnan H.* Fast portscan detection using sequential hypothesis testing // Proc. of the 2004 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 9—12, 2004. IEEE Computer Society. 2004. P. 211—225.

2. *Schechter S., Jung J., Berger A. W.* Fast detection of scanning worm infections // Proc. of the 7th Intern. Symposium on Recent Advances in Intrusion Detection, French Riviera, France, Sept. 2004. P. 59—81.

* Здесь и далее: Отделение информационных технологий и вычислительных систем — прежнее название Отделения нанотехнологий и информационных технологий РАН.

3. *Воронцов В. В., Котенко И. В.* Модели обнаружения и сдерживания сетевых червей на основе проактивного подхода // Материалы V Санкт-Петербург. межрегион. конф. „Информационная безопасность регионов России (ИБРР-2007)“, 23—25 окт. 2007 г. СПб., 2007. С. 47—48.

4. *Котенко И. В., Воронцов В. В.* Проактивный подход к обнаружению и сдерживанию сетевых червей // Тр. Междунар. науч.-техн. конф. „Интеллектуальные системы (AIS 07)“ и „Интеллектуальные САПР (CAD-2007)“. М.: Физматлит, 2007. Т. 2. С. 61—68.

Сведения об авторах

Виктор Васильевич Воронцов — СПИИРАН, научно-исследовательская группа компьютерной безопасности; E-mail: vorontsov@comsec.spb.ru

Игорь Витальевич Котенко — СПИИРАН, научно-исследовательская группа компьютерной безопасности; E-mail: ivkote@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 004.056

В. А. Десницкий, И. В. Котенко

МОДЕЛЬ ЗАЩИТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ МЕХАНИЗМА „УДАЛЕННОГО ДОВЕРИЯ“

Предложен подход к построению модели защиты программ от несанкционированных изменений и вмешательств с использованием механизма „удаленного доверия“. Рассмотрены основные составляющие элементы механизма и принципы его функционирования. Предложены два варианта реализации механизма замещения мобильного модуля на основе парадигмы аспектно-ориентированного программирования.

Ключевые слова: *защита программного обеспечения, удаленное доверие, верификация, мобильный модуль, динамическое замещение.*

Введение. Проблема защиты программного обеспечения от несанкционированных изменений и вмешательств („взлома“) представляет собой важное направление в области компьютерной безопасности и является актуальной, в первую очередь, для разработчиков программного обеспечения и владельцев авторских прав на него. Современные средства защиты от взлома должны гарантировать корректное функционирование программ в соответствии с заданными требованиями и ограничениями. В противоположность антивирусным средствам, главной целью которых является защита программного и аппаратного обеспечения от влияния программ, полученных извне и содержащих компрометирующий их вирусный код, механизмы защиты от взлома предназначены для защиты программ со стороны потенциально ненадежного программного окружения. Несанкционированные модификации могут совершаться злоумышленником посредством разнообразных программных инструментов, таких как декомпиляторы, дизассемблеры, отладчики, „прикрепляемые“ к выполняющейся программе, эмуляторы, дамперы и снифферы [1].

В настоящее время известно достаточно большое количество приемов защиты программного обеспечения от взлома. Однако используемые программные методики, реализующие защитные механизмы, являются достаточно слабыми и могут относительно легко быть нейтрализованы взломщиком (противником) за сравнительно небольшое время. Действительно, как показывает практика, любой программный метод, реализующий защиту программ, с большой долей вероятности может быть устранен человеком, обладающим определенными знаниями и владеющим необходимыми техническими средствами.

Таким образом, любой метод защиты может быть взломан за определенное время. Из этого следует, что для адекватной защиты программ от взлома необходимо создание принципиально новой схемы (модели). В частности, такая схема, помимо технологических сложностей нейтрализации защитных механизмов, должна учитывать решения, позволяющие накладывать временные ограничения на работу злоумышленника. В этой ситуации по истечении некоторого периода времени механизм защиты претерпевает изменения, сводящие на нет всю или некоторую часть усилий злоумышленника по осуществлению взлома.

Основным элементом рассматриваемой в настоящей статье модели защиты программного обеспечения от взлома является механизм „удаленного доверия“ [2]. Данный механизм предполагает осуществление защиты программных приложений посредством их программной верификации удаленным защищенным сервером в реальном времени. Рассмотрим более детально основные положения данного подхода.

Сущность механизма „удаленного доверия“. Основными объектами предлагаемой модели защиты являются клиентская программа, которая выполняется на ненадежном хосте и подлежит защите, и удаленный защищенный сервер, функционирующий на надежном хосте. Таким образом, предполагается, что защищаемая программа выполняется в среде, проверить надежность и достоверность которой не представляется возможным. Программа подвержена атакам со стороны пользователей, намеренных нарушить ее корректное выполнение. Для осуществления таких компрометирующих программу действий злоумышленник может использовать разнообразные инструменты взлома. Цель предлагаемого механизма защиты — гарантировать неизменность и корректность работы клиентской программы, работающей в потенциально враждебном окружении. В рамках рассматриваемой модели считается, что сервер, функционирующий в пределах надежного хоста, не может быть взломан. (Иначе говоря, защита хоста от удаленных сетевых атак представляет собой задачу, выходящую за рамки данной работы.) Для обеспечения функционирования механизма защиты необходимо наличие постоянного сетевого соединения между клиентской программой и удаленным надежным сервером.

Схема механизма „удаленного доверия“ приведена на рисунке.

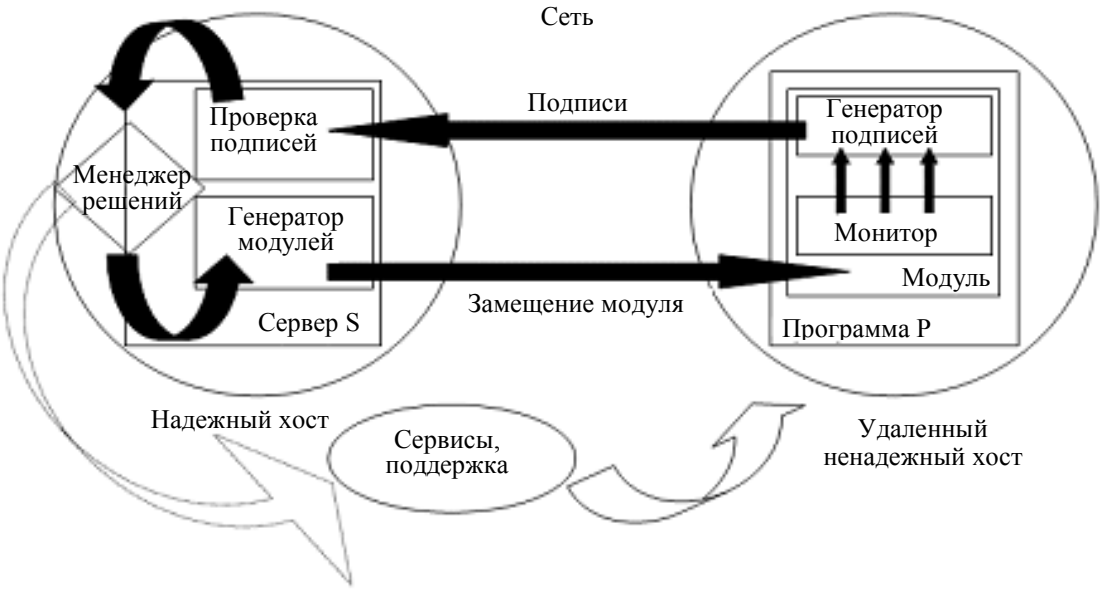


Схема механизма „удаленного доверия“

Один из важнейших принципов рассматриваемого подхода — внедрение в защищаемую программу специального переносимого (мобильного) программного модуля, основными элементами которого являются монитор и генератор подписей. В функцию монитора входит непрерывное и последовательное осуществление определенного набора проверок (верификаций)

клиентской программы во время ее выполнения, а также частично на стадии ее загрузки. Функции проверки включают верификацию бинарного кода, текущего состояния программы (в том числе, памяти, регистров и стека), используемых в программе структур данных, выполняющихся процессов, версий используемых библиотек, состояния операционной системы и пр. К проверкам также можно отнести выполнение специальных вычислений, чувствительных к изменениям программы. Возможны и другие виды верификаций. Важной особенностью переносимого модуля является то, что он не поставляется совместно с программой, а загружается с надежного хоста во время первой загрузки клиентского приложения и далее регулярно обновляется.

В соответствии с предлагаемым подходом при обнаружении вмешательств сервер прекращает предоставление данному клиенту любых сервисов, программных обновлений и других видов сопровождения. Основное условие применимости рассматриваемого механизма заключается в необходимости использования программой сетевых коммуникаций с другими хостами. Другими словами, программа либо должна осуществлять обмен данными с другими клиентами (как, например, программы IP-телефонии, чаты и т.п.), либо для ее работы необходимо наличие поддержки со стороны специализированного удаленного сервера (очевидным примером являются антивирусные программы, для корректной работы которых требуется периодическое обновление антивирусных баз).

Генерация и проверка цифровых подписей. При использовании рассматриваемого подхода клиент должен регулярно, с определенной периодичностью, отправлять надежному серверу результаты проверок, а также собранные монитором данные, характеризующие текущее состояние программы. Эта информация проходит стадию шифрования и затем отправляется надежному серверу в виде последовательности подписей. За процесс их формирования ответствен специальный компонент — генератор подписей, находящийся в составе загружаемого модуля. Главные задачи, связанные с отправляемыми серверу подписями, — однозначная идентификация клиента, который их создал, и выяснение различных характеристик, позволяющих определить корректность текущего состояния программы. Важным требованием к подписям является сложность их анализа потенциальным злоумышленником, не имеющим соответствующих криптографических ключей. Кроме того, стоит отметить требование об уникальности подписей (различие для каждого клиента и для каждого запуска программы), а также их неповторяемости (для предотвращения атак, основанных на повторной отправке ранее сформированных подписей).

Получив информацию от клиента, надежный сервер производит ее дешифрование, затем интерпретирует подписи и анализирует результаты верификаций и собранную клиентским монитором дополнительную информацию. За эти действия отвечает специальный располагающийся на сервере компонент, в функцию которого входит проверка подписей. После этого на основе полученных данных сервер принимает решение о том, было ли совершено вмешательство в работу клиентской программы или нет. Сервер считает программу некорректной, т.е. взломанной, если обнаруживает, что хотя бы одна из проверок завершилась с отрицательным результатом.

Динамическое замещение мобильного модуля. Важнейшей процедурой рассматриваемого подхода является замещение мобильного модуля. Этот механизм состоит в том, что во время выполнения программы происходит регулярная отправка сервером клиенту новой версии модуля, которая содержит обновленные версии функций верификации и сбора данных о программе. После доставки кода модуля на клиентскую машину осуществляется его установка в выполняющуюся программу без ее перезагрузки или приостановки. Замена модуля производится для повышения его устойчивости ко взлому. Возможные модификации кода модуля могут содержать изменения ключа шифрования, алгоритма работы модуля, его количественных и качественных характеристик и пр.

Для усложнения выполнения потенциальным злоумышленником атак на программу и, следовательно, увеличения времени их реализации программа может быть защищена также другими доступными методами и приемами защиты, например методами обфускации [3].

Период замещения модуля рассчитывается на основе оценивания времени, необходимого злоумышленнику для реализации взлома программы. Цель настоящего механизма — определить такой период замещения, чтобы не дать возможности противнику осуществить взлом программы в течение этого периода, когда будет функционировать определенная версия модуля. Основное требование к реализации механизма замещения состоит в том, чтобы время, затрачиваемое злоумышленником на взлом мобильного модуля, не оказывалось ниже определенного значения для всех последующих версий модуля при условии, что противник смог взломать предыдущие версии модуля.

В случае атаки на программу, если злоумышленник станет блокировать процедуру замещения модуля, надежный сервер обнаружит вмешательство, получив „старые“ подписи. Если же злоумышленник сможет заблокировать процесс отправки подписей серверу, то их отсутствие также будет трактоваться сервером как вмешательство в работу программы. В обоих случаях надежный сервер получает основание для принятия решения о том, что программа была атакована.

Методика расчета требуемого периода замещения может основываться на оценке вычислительной сложности выполнения атак на рассматриваемый механизм защиты, в том числе, сложности обратной разработки, де-обфускации, приемов преодоления применяемых дополнительных средств защиты. Дополнительными средствами оценки времени замещения могут служить экспериментальные и статистические методики, позволяющие учитывать человеческий фактор в процессе взлома программы. Такие оценки строятся на основе информации, предоставляемой экспертами в области безопасности, в том числе, на базе анализа действий специальных групп программистов (так называемых „красных команд“), которые позволяют на практике оценить стойкость механизмов защиты.

Дополнительным требованием, накладываемым на механизм замещения мобильного модуля, является взаимопроникновение кода модуля и кода клиентской программы, что позволяет обеспечить максимально возможное равномерное распределение кода модуля по коду программы. Данное требование непосредственно связано с целью повышения стойкости целевого приложения к атакам.

Распределение проверок между клиентом и сервером. Выполнение проверок может проходить как на ненадежном хосте, где выполняется клиентская программа, так и на надежном хосте, где функционирует сервер. Монитор клиентской программы, помимо того, что он выполняет функции верификации, также ответствен за сбор некоторой дополнительной информации, в частности входных данных для проверок, которые могут отправляться серверу. Поэтому в рамках сервера возможно частичное или полное повторение проверок, проведенных на стороне клиента. Возможно также выполнение отдельных проверок исключительно на сервере. В то же время следует учитывать, что полное перенесение проверок на сервер противоречит принципу, согласно которому программный код приложения должен выполняться в пределах клиентской машины. В противном случае рассматриваемый механизм „удаленного доверия“ становится слабо масштабируемым и утрачивает возможность обеспечивать корректную работу большого числа клиентов.

Помимо своей основной функции, монитор может также проверять корректность самого себя как части верифицируемой программы. Еще одно архитектурное улучшение основывается на реализации модуля с несколькими мониторами, каждый из которых способен верифицировать другие мониторы. Особенностью данного решения является необходимость одновременной модификации всех мониторов мобильного модуля для совершения противником атаки, удовлетворяющей его целям.

Реализация мобильного модуля на основе аспектно-ориентированного программирования. Для реализации механизма замещения мобильного модуля выбрана концепция аспектно-ориентированного программирования (АОП). АОП предоставляет возможность проводить изменения программы в реальном времени за счет внедрения небольших фрагментов кода — аспектов. АОП-машина [4] должна обеспечивать необходимый уровень взаимопроникновения кода модуля и остальной части кода программы, а также сокрытие кода модуля.

Для реализации мобильного модуля были выбраны АОП-машины, осуществляющие внедрение аспектов во время выполнения программы (runtime). При реализации этих АОП-машин используются средства отладчика, прикрепляемого к выполняющейся программе. Для выполнения аспектного кода отладчик приостанавливает работу программы, „исполняет“ нужный программный метод и затем возобновляет выполнение программы. Другой вариант — это JIT-подход (just-in-time), применяемый только в случае управляемого кода. В соответствии с JIT-подходом все программные изменения происходят, когда компилятор переводит объектный код в native (родной) код.

Заключение. Итак, представлен общий подход к построению модели защиты программ на основе механизма „удаленного доверия“. Главной целью предлагаемого подхода является обеспечение неизменности программы и правильности ее работы в потенциально враждебном окружении посредством обнаружения и уведомления доверенного хоста о несанкционированной модификации защищаемой программы. Реализация данного механизма позволяет гарантировать невозможность противнику использовать полноценным образом „взломанную“ копию программы. Перспективные направления исследований связаны с детальной разработкой отдельных элементов механизма „удаленного доверия“, их реализацией и анализом эффективности.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 07-01-00547), программы фундаментальных исследований ОИТВС РАН (контракт № 3.2/03), Фонда содействия отечественной науке, а также при частичной финансовой поддержке, осуществляемой в рамках проекта Евросоюза RE-TRUST (контракт № 021186-2).

СПИСОК ЛИТЕРАТУРЫ

1. *Atallah M., Bryant E., Stytz M.* A survey of anti-tamper technologies // J. of Defence Software Eng. 2004. Nov. P. 12—16.

2. *Десницкий В. А., Котенко И. В.* Модели удаленной аутентификации для защиты программ // Тр. Междунар. науч.-техн. конф. „Интеллектуальные системы (AIS'07)“ и „Интеллектуальные САПР (CAD-2007)“. М.: Физматлит, 2007. С. 43—50.

3. *Collberg C., Thomborson C.* Watermarking, tamper-proofing, and obfuscation-tools for software protection // IEEE Computer Society. 2001. Nov. P. 735—746.

4. *Десницкий В. А.* Реализация механизма замещения мобильного модуля на основе парадигмы аспектно-ориентированного программирования // Материалы V Межрегион. конф. „Информационная безопасность регионов России (ИБРР-2007)“. СПб., 2007. С. 49—50.

Сведения об авторах

Василий Алексеевич Десницкий — СПИИРАН, научно-исследовательская группа компьютерной безопасности; E-mail: desnitsky@comsec.spb.ru

Игорь Витальевич Котенко — СПИИРАН, научно-исследовательская группа компьютерной безопасности; E-mail: ivkote@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 004.056

Е. В. Сидельникова, А. В. Тишков, И. В. Котенко

ВЕРИФИКАЦИЯ ПОЛИТИК ФИЛЬТРАЦИИ С ПОМОЩЬЮ ИСЧИСЛЕНИЯ СОБЫТИЙ И АБДУКТИВНОГО ВЫВОДА

Рассматривается подход к верификации политик фильтрации с помощью абдуктивного вывода. Предлагается классификация аномалий в правилах таблицы доступа межсетевого экрана и способы их разрешения. Анализируются различные сценарии моделирования работы межсетевого экрана с помощью исчисления событий. Предлагаются способы применения абдуктивного поиска для нахождения аномалий в политике фильтрации и их разрешения на основе разбиения условий правил политики на непересекающиеся части.

Ключевые слова: *межсетевой экран, фильтрация трафика, абдуктивный вывод, аномалия фильтраций сетевого трафика.*

Введение. В задаче построения системы безопасности компьютерной сети существенную роль играет политика фильтрации трафика. За пропуск и блокировку пакетов отвечают межсетевые экраны (МЭ). Обработка потоков трафика через МЭ осуществляется на основе правил политики фильтрации, представляемых как строки таблицы доступа межсетевого экрана. Наличие избыточных условий в правилах, а также невыполнимых правил неоправданно замедляет работу МЭ, а следовательно, и всей сети. Выявление и разрешение подобных аномалий особенно актуально для больших компьютерных сетей.

Задача верификации правил в таблице доступа МЭ рассматривалась многими авторами. В работе [1] определены типы отношений между правилами фильтрации и на их основе представлена классификация аномалий, возникающих на МЭ. В работе [2] описана система „Firewall Policy Advisor“, которая осуществляет поиск аномалий фильтрации и затем редактирует таблицу доступа МЭ. Эта система определяет аномалии внутри одного и между различными МЭ, но без построения сетевой модели и применения ее к политике безопасности. Для обнаружения и разрешения аномалий на МЭ используется аргументационная логика и абдуктивный вывод [3, 4]. Применение исчисления событий к политикам авторизации и политикам обязательного выполнения рассматривается в работе [5].

В настоящей статье предлагается новый подход к верификации политик фильтрации МЭ. Особенность предлагаемого подхода по сравнению с другими, основанными на исчислении событий и абдуктивном выводе (рассматриваемыми, в частности, в работах [3, 5]), заключается в следующем. Моделирование действий МЭ осуществляется с помощью исчисления событий (ИС) [6]. После построения предметно-зависимых аксиом и формул ИС, выражающих политики фильтрации и действия МЭ, процедура абдуктивного вывода по заданному номеру правила таблицы доступа МЭ получает сценарий (последовательность событий), который приводит к „срабатыванию“ этого правила. Анализ полученного сценария позволяет найти „затененные“ правила (или части правил) и удалить их из таблицы доступа МЭ. В общем случае предлагаемый метод может быть применен для нахождения и разрешения различных аномалий фильтрации, а также для улучшения конфигурации МЭ.

Аномалии в правилах фильтрации и стратегии их разрешения. Правила фильтрации разрешают или запрещают пересылку пакетов с определенным адресом и портом источника и получателя трафика для заданного протокола. Упорядоченные правила составляют таблицу доступа МЭ. Каждое правило в таблице представляет собой тройку $< O, C, A >$, где O — номер правила в таблице доступа; C — условие, при котором пересылка пакета

разрешается или запрещается; A — действие МЭ для данного правила, т.е. разрешение или запрещение пересылки пакета в случае, если для пакета выполнено условие C.

Аномалия в таблице доступа возникает, если условия хотя бы двух правил пересекаются. Пусть имеются два правила $< O_1, C_1, A_1 >$ и $< O_2, C_2, A_2 >$, где $O_1 < O_2$. В зависимости от типа пересечения C_1 и C_2, а также расположения правил в таблице можно выделить следующие типы аномалий: *затенение*, если $C_2 \subseteq C_1$; *обобщение*, если $C_1 \subseteq C_2$; *корреляция*, если $C_1 \cap C_2 \neq \varnothing$, $C_1 \setminus C_2 \neq \varnothing$ и $C_2 \setminus C_1 \neq \varnothing$.

Для данных типов аномалий известны различные стратегии разрешения:

— *преимущество разрешения* — устанавливает приоритет разрешающему правилу и применима к аномалии, в которой $A_1 \neq A_2$;

— *преимущество запрещения* — устанавливает приоритет запрещающему правилу и применима к аномалии, в которой $A_1 \neq A_2$;

— *преимущество более специфичного* — устанавливает приоритет правилу с меньшей областью действия условия, применима только к аномалиям затенения;

— *преимущество менее специфичного* — устанавливает приоритет правилу с большей областью действия условия, применима только к аномалиям обобщения;

— *удаление невыполнимого* — удаляет правило, которое никогда не выполняется, применима только для аномалии затенения;

— *разбиение* — разбивает условия правил на непересекающиеся части, добавляя новые правила в таблицу доступа МЭ или удаляя прежние.

Действие первых четырех стратегий заключается в изменении их порядка в таблице доступа МЭ, что в общем случае приводит к изменению поведения МЭ. Рассмотрим последние две стратегии разрешения, не меняющие поведения МЭ.

Аксиоматика исчисления событий. Исчисление событий [6] формализует общий принцип инерции: свойство окружающего мира, называемое в терминологии ИС флюентой, изменяется только под воздействием событий и остается неизменными в промежутках между ними. Флюента реализуется в некоторый момент времени, если она была инициирована событием в более ранний момент времени и не была приостановлена каким-либо другим событием между этими двумя моментами. Аналогично флюента не реализуется в некоторый момент времени, если она была ранее приостановлена и не была инициирована после. ИС является языком первого порядка, в котором флюенты, события и моменты времени выступают в качестве сортов.

Общий принцип инерции выражается при помощи следующих предикатов: предикат $\mathrm{holds\_at}(F, T)$ верный, если флюента F реализуется в момент времени T; предикат $\mathrm{happens}(A, T)$ верный, если в момент времени T произошло событие A; предикат $\mathrm{initially}(F)$ верный, если флюента F реализуется в начальный момент времени; предикат $\mathrm{terminates}(A, F, T)$ верный, если событие A приостановило флюенту F в момент времени T; предикат $\mathrm{initiates}(A, F, T)$ верный, если событие A инициировало флюенту F в момент времени T; предикат $\mathrm{clipped}(T_1, F, T_2)$ верный, если между моментами времени T_1 и T_2 флюента F была приостановлена некоторым событием; предикат $\mathrm{declipped}(T_1, F_1, T_2)$ верный, если между моментами времени T_1 и T_2 флюента F была инициирована некоторым событием; вспомогательный предикат $\mathrm{holds}(T_1, F, T_2)$ верный, если флюента F выполняется между моментами времени T_1 и T_2 при условии, что в это время не происходило событий, влияющих на ее реализацию; предикат $\mathrm{precondition}(A, T)$ верный, если в момент времени T выполнены условия для того, чтобы произошло событие A.

При помощи данных предикатов и отрицания флюенты neg(F) можно записать набор из предметно-независимых аксиом ИС. В настоящей статье аксиомы записывались с помощью языка скриптов, предоставляемого средствами абдуктивного вывода CIFF [7].

Моделирование поведения МЭ. Моделирование действий МЭ задается предметно-зависимой аксиоматикой ИС. Рассмотрим поведение МЭ (с правилами $< 0, C_0, A_0 >$, $< 1, C_1, A_1 >$, …, $< N, C_N, A_N >$), на который приходит запрос для пакета P со свойствами, определяющими протокол, адрес и порт источника и получателя. Пусть в таблице доступа МЭ свойства пакета P удовлетворяют условию C_K, причем для любого $i < K$ свойства пакета P не удовлетворяют C_i, т.е. для данного пакета выполняется K-я строка таблицы доступа МЭ.

Схема моделирования поведения МЭ в терминологии ИС показана на рисунке. Событие request(P) происходит, когда на МЭ поступает запрос на обработку пакета P. Это событие инициирует флюенту checking(P), которая выполняется, пока МЭ обрабатывает данный запрос, и флюенту checking_rule(0), которая выполняется, пока проверяется правило с номером 0 в таблице межсетевого экрана.

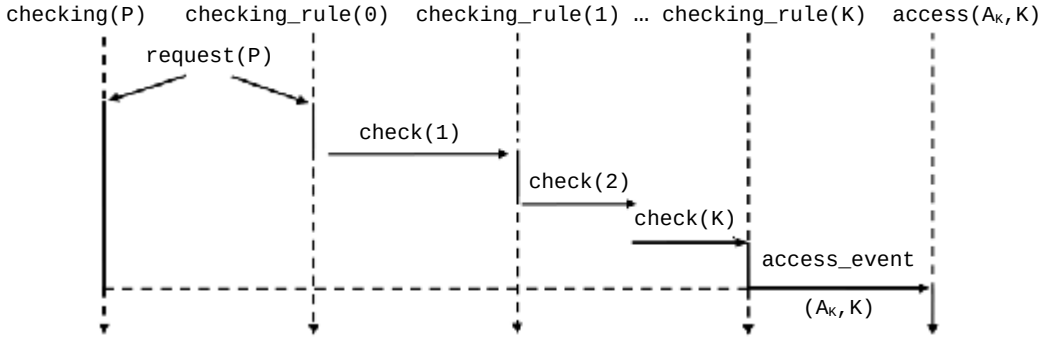


Схема моделирования поведения межсетевого экрана

Если пакет не соответствует условию 0-го правила, то событие check(1) инициирует флюенту checking_rule(1) и терминирует флюенту checking_rule(0), и так далее, пока не начнет реализовываться флюента checking_rule(K), т.е. МЭ не перейдет к проверке K-й строки. Тогда событие access_event(A_K, K) инициирует флюенту access(A_K, K) и терминирует флюенту checking(P), т.е. МЭ заканчивает работу и выдает результат A_K, где A_K принимает значения allow или deny (разрешить или запретить передачу пакета соответственно).

Поиск и разрешение аномалий с помощью абдуктивного вывода. После формализации правил работы МЭ на основе средств абдуктивного вывода CIFF[7] можно выяснить, какие правила будут выполняться, когда на МЭ придет запрос на пересылку пакета. Рассмотрим пример таблицы доступа МЭ.

Номер правила	Протокол	Адрес источника	Порт источника	Адрес получателя	Порт получателя	Действие
0	tcp	140.192.37.20	80	*.*.*.*	Любой	Разрешить
1	tcp	140.192.37.*	80	*.*.*.*	Любой	Запретить
2	tcp	140.192.37.30	80	161.120.33.40	Любой	Разрешить

В таблице правило под номером 2 не сработает никогда, так как оно „затенено“ правилом 1. Для выявления данной аномалии напишем следующий запрос:

?- run_ciff([*policies*], [holds_at(access(*Action*, 2), T)], *Answer*),

где *policies* — это файл, содержащий аксиоматику, приведенную выше; holds_at(access(*Action*, 2) — предикат, задающий выполнение флюенты access(*Action*, 2)

в момент времени T, т.е. регламентирующий тот факт, что второе правило МЭ будет выполнено в некоторый момент времени; *Answer* — ответ, содержащий абдуцирумые предикаты и соотношения между ними.

В рассматриваемом случае абдуцируемым предикатом является предикат happens/2. В файле *policies* такой предикат указывается в виде abducible(happens(_,_)). Целью абдуктивного алгоритма является нахождение сценария или последовательности событий в виде набора пар $\{(e_i, t_i)\}_{1 \le i \le N}$, где e_i — событие, произошедшее в момент времени t_i, и $t_i \le t_j$ для любых $i < j$.

Очевидно, что для рассматриваемого запроса невозможно найти ни одного сценария, так как любой пакет, удовлетворяющий условию правила 2 таблицы доступа МЭ, будет запрещен правилом 1.

Предложим два способа разбиения правил на непересекающиеся множества для аномалий обобщения и корреляции.

Первый способ разбиения правил напрямую использует предложенную в работе аксиоматику для моделирования поведения МЭ.

Рассмотрим следующие запросы к МЭ, заданному в таблице.

?- run_ciff([*policies*], [holds_at(access(*Action*, 0), *T*)], *Answer*), (1)

?- run_ciff([*policies*], [holds_at(access(*Action*, 1), *T*)], *Answer*). (2)

Результат, выдаваемый алгоритмом CIFF на запрос (1), будет, во-первых, содержать подстановку для переменной *Answer*, состоящую из конъюнкции предикатов

$$\text{happens (request (condition (}tcp\text{, s\_ip (140, 192, 37, 20), 80, d\_ip(}\_B, \_C, \_D, \_E), \_F)), \_G),$$
$$\text{happens (access\_event (}allow, 0), \_A)$$

и неравенств $\_G < \_A$, $\_G < T$ и $\_A < T$. Также в подстановку для переменной *Answer* входит список $[\_A = 1, \_G = 0]$, состоящий из возможных значений моментов времени. Во-вторых, в ответ для данного запроса входит подстановка для остальных переменных: $Action = allow$, $T = 2...200$. Данный ответ означает следующее: для того чтобы было выполнено правило 0 в таблице доступа МЭ, должен прийти tcp-пакет с адресом источника 140.192.37.20, портом источника 80, произвольным портом получателя и произвольным адресом получателя.

Ответ на запрос (2) содержит абдуцируемые предикаты

$$\text{happens(request(condition(}tcp\text{,s\_ip(140,192,37,}\_C\text{),80,d\_ip(}\_D, \_E, \_F, \_G), \_H)), \_I),$$
$$\text{happens(check(1),}\_B), \ \text{happens(access\_event(}deny\text{,1), }\_A),$$

причем $\_A < \_B < \_C < T$, а также ограничение $\_C \ne 20$. В этом ограничении заключается отличие правила 1 от запросов, которые „пройдут“ по правилу 2.

Также следует заметить, что между правилами 1 и 2 таблицы существует аномалия обобщения, которую можно разрешить, добавив данное ограничение к правилу 1:

$$\text{rule}(N, C, Action) :- Action = deny, N \# = 1,$$
$$C = \text{condition}(tcp, \text{s\_ip}(140,192,37,W_1), 80, \text{d\_ip}(X_2, Y_2, Z_2, W_2), P_2),$$

где # обозначает сравнение значений. Такое изменение соответствует применению стратегии разбиения к данной аномалии. Делая запросы последовательно ко всем правилам и сравнивая результаты с исходными правилами, можно составить новый набор правил с непересекающимися условиями.

Второй подход для поиска аномалий основывается на следующем рассуждении. Если взять произвольное правило, а затем исключить его из таблицы доступа, в которой нет пере-

секающихся правил, то для множества пакетов, для которых оно выполняется (найденное при помощи абдуктивной процедуры), не найдется правила в новой таблице доступа. Если же такое правило найдется, то между ним и первым правилом будет существовать аномалия. Таким образом, можно последовательно находить пары правил с аномалиями, и их разрешение оставить на усмотрение пользователя.

Заключение. В настоящей статье представлен подход к верификации политик фильтрации межсетевого экрана. Использование предложенной аксиоматики исчисления событий при моделировании МЭ позволяет решать задачу анализа его работы и настройки. В предлагаемом подходе выделены прямая и обратная *задачи анализа работы МЭ*: прямая задача — по свойствам трафика, передаваемого на МЭ, вычислить действие типа allow/deny; обратная задача — по номеру сработавшего правила определить свойства трафика. Решение обратной задачи с использованием абдуктивного вывода, представленное на базе библиотеки CIFF для продукта SICStus Prolog, было программно реализовано. Решение прямой задачи предполагается реализовать на SICStus Prolog с помощью применения дедуктивного вывода. *Задача конфигурирования МЭ* заключается в определении на основе свойств трафика и желаемого отклика МЭ такого набора правил, который, например, содержит минимальное их количество или имеет минимальное количество пересечений условий. Задача конфигурирования по заданным ограничениям на прохождение трафика также может быть решена с использованием абдуктивного вывода и предложенной аксиоматики исчисления событий. В настоящее время эта задача находится на стадии реализации программного прототипа.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект № 07-01-00547), программы фундаментальных исследований ОИТВС РАН (контракт № 3.2/03), Фонда содействия отечественной науке, а также при частичной финансовой поддержке по проекту Евросоюза RE-TRUST (контракт № 021186-2).

СПИСОК ЛИТЕРАТУРЫ

1. *Al-Shaer E., Hamed H., Boutaba R., Hasan M.* Conflict classification and analysis of distributed firewall policies // J. on Selected Areas in Communications. 2005. Vol. 23, N. 10. P. 2069—2084.
2. *Al-Shaer E., Hamed H.* Firewall policy advisor for anomaly discovery and rule editing // IEEE/IFIP Integrated Management. 2003. 14 p.
3. *Bandara A. K., Kakas A. S., Lupu E. C., Russo A.* Using argumentation logic for firewall policy specification and analysis // 17th IFIP/IEEE Intern. Workshop on Distributed Systems. 2006. P. 185—196.
4. GORGIAS. Argumentation and Abduction [Электронный ресурс]: <http://www2.cs.ucy.ac.cy/ ~nkd/gorgias/>.
5. *Bandara A. K., Lupu E. C., Russo A.* Using event calculus to formalise policy specification and analysis // IEEE Workshop on Policies for Distributed Systems and Networks. 2003. P. 26—42.
6. *Kowalski R. A., Sergot M. J.* A logic-based calculus of events // New Generation Computing. 1986. N. 4. P. 67—95.
7. *Endriss U., Mancarella P., Sadri F.* et al. The CIFF proof procedure: definition and soundness results // Technical Rep. Department of Computing, Imperial College London. 2004. Vol. 2. 17 p.

Сведения об авторах

Екатерина Викторовна Сидельникова — СПИИРАН, научно-исследовательская группа компьютерной безопасности; E-mail: sidelnikova@comsec.spb.ru

Артем Валерьевич Тишков — СПИИРАН, исследовательская группа информационных технологий в образовании; E-mail: avt@iias.spb.su

Игорь Витальевич Котенко — СПИИРАН, научно-исследовательская группа компьютерной безопасности; E-mail: ivkote@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 004.41

А. Ю. Подъячев, А. Ю. Атисков, С. В. Перминов

ТЕСТИРОВАНИЕ ПРОЦЕССА ТРАНСФОРМАЦИИ ФУНКЦИОНАЛЬНЫХ МОДЕЛЕЙ НА ОСНОВЕ ГИБРИДНЫХ МЕТОДОВ

Обсуждается задача трансформации формальных моделей IDEF0 в диаграммы классов UML с использованием гибридных методов, включающих онтологическое проектирование правил трансляции. Рассматривается механизм создания правил трансформации и тестирования на соответствие полученной и исходной моделей.

Ключевые слова: *оценка качества, модельное тестирование, трансляция моделей, онтологические правила.*

Введение. С развитием стандартизации и интенсификацией процессов глобализации методов и средств разработки программных систем средства для кроссметодологических трансформаций моделей становятся все более востребованными. Этим обусловлено создание автоматизированной системы, предназначенной для решения задачи трансформации модели из нотации IDEF0 в диаграмму классов UML [1].

Разработка систем такого типа связана с проблемой корректности описания функционального поведения при трансляции моделей. Несмотря на то что модели представляют собой ориентированные взвешенные графы, использование последних для решения данной задачи осложнено неоднородностью методов их оценки. Предлагаемый в настоящей статье подход связан с использованием существующих средств тестирования и оценки качества моделей в новом контексте для определения полноты трансляции, а также сложности полученного графа и вычисления его метрических оценок.

Задачи трансформации моделей. При решении задачи построения автоматизированного средства трансформации моделей использовались три основанные на системе правил подхода — от самого простого до гибридного — с возможностью их адаптации к исходным данным [1—3]. Свойства этих подходов в соответствии с классификацией, предложенной в работе [4], приведены в таблице.

Подход на основе прямой манипуляции обеспечивает полный контроль над описанием правил трансформации и позволяет строить правила любой сложности, но при этом требуется изменять программный код приложения и выполнять его перекомпиляцию. Кроме того, данный подход не предполагает использование механизмов формального описания правил и элементов моделей, что в долгосрочной перспективе делает его непрактичным.

Применение структурного подхода позволяет дифференцировать правила трансформации и основную программу. Такой подход предоставляет большую свободу пользователю при проведении трансформации, однако требует знания языка программирования Java для определения даже простых правил.

Использование гибридного подхода приводит к построению метода трансформации, который позволяет реализовать модульную программную систему. В качестве основных ее свойств можно выделить следующие: независимость от синтаксиса входной и выходной нотаций модели; использование формализованной семантики для определения нотаций проектирования и правил трансформации, реализуемой посредством языков RDF, OWL [3] и SPARQL [5]; контроль за стратегией выполнения правил со стороны пользователя; адаптивность системы к входным и выходным данным при использовании независимых семантических анализаторов данных.

Форма представления правил	Подход		
	прямая манипуляция	структурный	гибридный
Переменные правил	Синтаксически типизированы	Синтаксически типизированы	Семантически типизированы
Представление правил	Строковое	Строковое	Графовое
Синтаксис правил	Текстовый	Текстовый	Абстрактный
Типизация правил	Нетипизированные		
Логика правил	Императивная	Императивная	Декларативная
Ограничение области применения	Есть	Есть	Нет
Редактирование модели	В выходной целевой модели (отсутствие правки в процессе применения правил)		
Параметризация правил	Есть		
Стратегия правил	Детерминистическая	Детерминистическая	Недетерминистическая
Схема планирования	Неявная	Явная	Явная
Планирование применения правил	Внутреннее	Внешнее	Внешнее
Планирование итераций	Рекурсии, циклы	Рекурсии, циклы	Одноэлементное
Планирование фаз трансформации	Есть		
Модули правил			
Синтаксическое разделение			
Повторное использование правил			
Организационная структура правил	По исходным данным	По исходным данным	Независима
Направленность правил	Однонаправленные	Однонаправленные	Возможна двунаправленность
Связи между исходными и целевыми элементами	Нет	Нет	Есть

Формализация правил трансформации. Рассмотрим формализацию семантики на примере правила, состоящего из двух утверждений, в первом из которых говорится о том, что определенные дуги исходного графа являются объектами класса, а во втором — о том, что блок исходного графа — это метод (структура) класса полученной модели.

Сначала формализуем преобразование для первого утверждения. Так как задача преобразования сводится к трансформации одного RDF-формата данных (описывающего IDEF0-диаграмму) в другой (описывающий UML-диаграмму), используем стандартизованный язык запросов SPARQL [5] для извлечения данных из IDEF0-диаграммы. RDF-идентификаторы при преобразовании будут сохраняться. Предварительно определим XML-префикс, который будет использоваться во всех последующих формализмах:

```
PREFIX tf: <spiiras.transform>.
```

Часть SPARQL-запроса, описывающего объект класса, выделим в отдельную константу, содержащую текст для использования и в других запросах:

```
%class =
?class a tf:Mechanism .
 _:class a tf:Arrow .
_:class tf:hasArrowEnd ?class
```

Запрос:

```
SELECT ?class ?name
WHERE {
%class .
_:class tf:name ?name }
```

Для приведения запроса к окончательному (стандартизованному) виду достаточно произвести в нем предварительную обработку с заменой константы `%class` ее значением и добавлением информации об используемом префиксе.

Другая часть преобразования состоит в генерации выходных RDF-данных для UML-диаграммы. Для простоты редактирования и восприятия правил построения выходных данных используем нестандартную нотацию (основанную на примитивах языка SPARQL), которая описывает логические тройки, создаваемые для каждого элемента массива, получаемого как результат работы SPARQL-процессора:

```
?class a tf:Class .
?class tf:name ?name
```

Таким же образом формализуем преобразование для второго утверждения.

Выделяем константу

```
%method =%class .
?method a tf:Block .
?class tf:atBlock ?method,
```

тогда запрос будет сформирован в следующем виде:

```
SELECT ?class ?method ?name
WHERE {%method .
?method tf:name ?name }.
```

Результирующие данные в этом случае будут соответствовать следующему выражению:

```
?class a tf:Class .
?method a tf:Method .
?class tf:method ?method .
?method tf:name ?name.
```

Получаемые независимые друг от друга декларативные правила преобразования с формализованной семантикой могут выполняться в любом порядке. Как одно из следствий использования декларативного подхода при построении правил в выходных данных возможно появление копий логических троек, которые необходимо удалить на любом из этапов преобразования. Подобным образом можно формализовать и другие правила преобразования диаграмм, состоящие из утверждений. Таким образом, можно говорить, что задача формализации правил, т.е. формального описания их семантики, решена.

Оценка формальной модели. Существует достаточное количество метрик, которые в той или иной степени могут характеризовать IDEF0-модели. Часть этих метрических оценок применима также и к диаграммам классов UML [6, 7]. Взаимосвязь метрических показателей может быть использована для проверки соответствия результата гибридной трансформации модели IDEF0 в диаграмму классов UML. Рассмотрим такую взаимосвязь на примере примитивной метрики покрытия элементов исходной спецификации системы**.** Тестовый набор элементов модели удовлетворяет критерию полного покрытия, если при его проверке каждому элементу находится соответствие в исходной спецификации программы, по крайне мере, один раз.

Для отображения результатов проверки модели достаточно хорошо подходят методы проверки пустого и ложного покрытий. Пусть F — диаграмма классов UML; φ — спецификация IDEF0, удовлетворяющая F; F' — измененная диаграмма UML. Если F' не удовлетворяет φ, то можно однозначно утверждать, что спецификация φ покрыта ошибочно. При F', удовлетворяющем φ, можно также говорить о бессодержательном покрытии спецификации φ. Таким образом, проверяем, удовлетворяет ли спецификации измененная схема полученной модели, в случае если спецификация также подвергалась изменению.

Для точного определения результатов тестирования моделей необходимо, помимо использования метода оценки формальной модели, применить метод оценки ошибочного по-

крытия элементов спецификации. Алгоритм этого метода позволяет произвести вычисления, осуществляемые путем подмены значений переменных. Подмена значений происходит в наборах выходных параметров методов классов UML-диаграммы. Подробное описание этого алгоритма изложено в работе [8]. Для символьных проверок соответствия и оценки метрических показателей подходит предложенное выше представление элементов диаграмм обеих нотаций в стандарте RDF [3].

Заключение. Для решения задачи формализации правил трансформации был задействован пакет технологий, предложенных в рамках пирамиды стандартов „Semantic Web“ [9]. Это является экспериментальным подтверждением предположения о том, что методика семантического поиска пригодна для анализа и трансформации не только данных, представленных в сети Интернет, но и для любых других данных, если последние можно формализовать для представления в стандарте RDF. Суть предложенной в настоящей статье методики сводится к извлечению данных и их контекста из исходного документа, формализации представления извлеченной информации в виде семантической сети и последующему анализу полученных данных.

С другой стороны, говоря о технологии тестирования, необходимо отметить, что она может применяться не только как прикладной инструмент для автоматизации тестирования, но и как инструмент для исследования методов построения моделей программного обеспечения в различных нотациях и их дальнейшего использования в качестве спецификаций для создания тестовых сценариев [10]. Перспективное развитие методов метрической оценки данных моделей предоставит возможность контролировать сложность, надежность и качество как исходной спецификации программы, так и последующих реализаций программных комплексов. По оценкам схемы построения тестового сценария можно сделать выводы и дать рекомендации относительно улучшения кода либо пересмотра спецификации при наличии проблем с надежностью системы. Предложенный алгоритм тестирования дает возможность абстрагироваться от конкретных описаний метамоделей и, в частности, позволяет использовать IDEF0-описание в качестве тестовой спецификации.

Данная технология использована при выполнении проекта 1.3 целевой программы Президиума РАН „Информатизация“, а также комплексного междисциплинарного проекта „Разработка портала и баз данных Президиума Санкт-Петербургского научного центра РАН“.

СПИСОК ЛИТЕРАТУРЫ

1. *Атисков А. Ю., Воробьев В. И.* Автоматизированная система трансформации диаграмм бизнес-процессов в диаграммы классов // Тр. СПИИРАН. СПб.: Наука, 2006. Вып. 3, т. 2. С. 146—155.

2. *Атисков А. Ю., Воробьев В. И.* Адаптивная система трансформации диаграмм бизнес-процессов в диаграммы классов // Вестн. гражданских инженеров. СПб.: СПбГАСУ, 2007. № 1(10). С. 83—88.

3. *Атисков А. Ю., Перминов С. В.* Гибридная адаптивная технология проектирования бизнес-процессов // Тр. СПИИРАН. СПб.: Наука, 2007. Вып. 4. С. 184—192.

4. The Open Group. ADL 2.0 Translation System [Электронный ресурс]: <http://adl.opengroup.org/> (по состоянию на 10.03.2007).

5. SPARQL Query Language for RDF. W3C Recommendation 15 January 2008 [Электронный ресурс]: <http://www.w3.org/TR/rdf-sparql-query/> (по состоянию на 14.04.2008).

6. *Подъячев А., Афанасьев С.* Тестирование трансляции формальных моделей // Тр. СПИИРАН. СПб.: Наука, 2006. Вып. 3, т. 2. С. 156—161.

7. *Подъячев А.* Построение TTCN тестов на основе UML диаграмм // Там же. 2007. Вып. 4. С. 155—162.

8. *Подъячев А.* Использование метода покрытий при верификации моделей IDEF-0 // Там же. 2007. Вып. 5. С. 275—283.

9. *Berners-Lee T., Hendler J, Lassila O.* The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities // Scientific Amer. 2001. N. 5. P. 34—43.

10. *Adel K., Helsen S.* Classification of model transformation approaches // OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture. Univ. of Waterloo, Canada, 2003.

Сведения об авторах

Алексей Юрьевич Подъячев — СПИИРАН, лаборатория вычислительных систем и проблем защиты информации; E-mail: Alexey.Podjachev@quest.com

Алексей Юрьевич Атисков — СПИИРАН, лаборатория вычислительных систем и проблем защиты информации; E-mail: atiskov@gmail.ru

Сергей Владимирович Перминов — СПИИРАН, лаборатория вычислительных систем и проблем защиты информации; E-mail: sv.perminov@gmail.com

Поступила в редакцию
06.05.08 г.

МЕТОДЫ И СРЕДСТВА ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

УДК 004.93

А. Л. Ронжин, А. А. Карпов

СРАВНЕНИЕ МЕТОДОВ ЛОКАЛИЗАЦИИ ПОЛЬЗОВАТЕЛЯ МНОГОМОДАЛЬНОЙ СИСТЕМЫ ПО ЕГО РЕЧИ

Рассматривается проблема дистанционной записи и распознавания речи для задачи голосового взаимодействия с автоматической информационно-справочной системой в условиях акустических шумов. Благодаря пространственной локализации источников звука система воспринимает и анализирует звуки, исходящие из узкой области пространства в рабочей зоне. Приведены результаты тестирования трех методов определения направления к источнику звука с использованием различных схем массивов микрофонов.

Ключевые слова: *дистанционное распознавание речи, массив микрофонов, многомодальный интерфейс.*

Введение. Реализация долгосрочной цели — создания всепроникающей компьютерной технологии (Ubiquitous Computing) — требует также решений в области многоканального акустического анализа и создания робастных методов локализации пользователя в пространстве, слежения за его перемещением, определения речевой активности и дистанционного распознавания речи [1]. Сложность последней проблемы заключается, прежде всего, в том, что необходимо автоматически осуществлять мониторинг за источником полезного сигнала (речь пользователя) и источниками шумов в реальных реверберационных условиях без ограничений на число одновременно функционирующих источников звука [2].

За последние десятилетия были достигнуты значительные результаты в области дистанционного распознавания речи, тем не менее высокая производительность систем распознавания обеспечивается только для небольших словарей, при одинаковых окружающих акустических условиях, в которых производится обучение и тестирование, а положение пользователя, ориентация головы и стиль речи остаются неизменными в течение диалога [3]. В таких приложениях применение одного или нескольких массивов микрофонов становится особенно эффективным благодаря их способности оценивать пространственное положение источников звука. Эффективность массива микрофонов существенно зависит от его геометрии и алгоритмов формирования луча, которые используются для комбинирования сигналов, поступающих с нескольких микрофонов, а также от других методов подавления и фильтрации шумов, включенных в полную схему цифровой обработки многомерного сигнала.

Существующие методы определения положения источника звука можно разделить на две группы: параметрические и непараметрические. Параметрические подходы, основанные на формировании луча (beamforming) или вероятностных методах (maximum likelihood approaches), определяют пространственную функцию вероятности для каждой точки

пространства [4]. Такая функция может иметь несколько локальных максимумов. Анализ пространства для всех локальных максимумов этой функции является длительным процессом.

Непараметрические методы, также известные как методы анализа сигналов подпространства с высокой разрешающей способностью или методы анализа собственных чисел (eigen analysis), не зависят от подобной функции. Например, хорошо известные алгоритмы MUSIC [5] и ESPRIT [6], не использующие параметрические методы, обеспечивают высокую разрешающую способность.

В простых приложениях определение положения пользователя в пространстве осуществляются путем измерения времени задержки между сигналами, записанными двумя или более микрофонами. В большинстве приложений используются методы обобщенной функции взаимной корреляции (General Cross Correlation — GCC) [7] или обработки фазы сигналов [8, 9] для оценивания задержки прихода звуковой волны. В таких методиках положение пользователя определяется с помощью набора оценок задержек, вычисленных путем сравнения сигналов, записанных с разных микрофонов. Основной недостаток перечисленных методов проявляется в условиях высокой реверберации, когда происходит множественное отражение звуковых волн от стен помещения, и основной сигнал перемешивается с его отраженными копиями. Также следует отметить, что проблема локализации нескольких источников звука стала исследоваться совсем недавно, хотя такая ситуация наиболее свойственна для реальных условий.

Методы оценки времени задержки сигналов. Для моделирования сигнала, излучаемого удаленным источником в условиях шумов и записанного несколькими разнесенными в пространстве микрофонами, обычно используется следующее выражение [7]:

$$x_i(n) = \alpha_i s(n - \tau_i) + b_i(n), \qquad (1)$$

где $x_i(n)$ — сигнал, записанный i-м микрофоном; α_i — коэффициент ослабления сигнала при распространении в воздухе; τ_i — время прохождения звуковой волны от источника $s(n)$ до i-го микрофона; $b_i(n)$ — аддитивный шум i-го микрофона.

Предполагается, что $s(n), b_i(n)$ — независимые случайные гауссовские процессы. Тогда относительная задержка между сигналами, записанными двумя микрофонами, определяется как разность между временем прохождения волны от источника до первого и второго микрофонов: $\tau_{12} = \tau_1 - \tau_2$.

Такая модель является идеальной, и если записать выражение (1) в частотной области:

$$X_i(f) = \alpha_i S(f) e^{-j2\pi f \tau_i} + B_i(f),$$

затем вычислить знак (комплексный) взаимного спектра $G_{x_1x_2}(f)$ между $X_1(f)$ и $X_2(f)$

$$\operatorname{sgn}[G_{x_1x_2}(f)] = \operatorname{sgn}[E\{X_1(f)\overline{X_2}(f)\}] = e^{-j2\pi f \tau_{12}}, \qquad (2)$$

где $\operatorname{sgn}(z) = z/|z|$, $E\{.\}$ — математическое ожидание, $\overline{(\cdot)}$ обозначает комплексно-сопряженное число, то легко увидеть, что обратное преобразование Фурье от выражения (2) даст явный максимум в точке, соответствующей задержке τ_{12} между сигналами.

К сожалению, для реальных акустических условий, когда необходимо учитывать эффект реверберации, различие в характеристиках микрофонов и направленность шумов, идеальная модель сигнала не всегда подходит. В этом случае применяют более сложные модели, учитывающие импульсные характеристики между источником и микрофоном [10].

Более эффективным способом оценки задержки является метод обобщенной функции взаимной корреляции GCC, который определяется следующим выражением [7]:

$$\hat{\tau}_{\mathrm{GCC}} = \arg\max_n \sum_{k=0}^{N-1} \psi_G(k) G_{x_1x_2}(k) e^{\frac{j2\pi nk}{N}},$$

где $G_{x_1x_2}(k) = X_1(k)\overline{X}_2(k)$ — взаимный спектр, $\psi_G(k)$ — некоторая весовая функция, N — длина анализируемого сегмента сигнала.

Метод GCC является более робастным по сравнению с простой автокорреляционной функцией, поскольку основан на предварительной фильтрации входных сигналов в некотором конечном окне, что позволяет избежать смешивания сигналов от различных источников и устранить влияние реверберации [1]. Недостаток данного метода заключается в том, что функция взаимной корреляции обычно имеет довольно размытый максимум, в результате невозможно достичь высокой точности в оценке задержки.

Для повышения производительности метода GCC применяют различные весовые функции, которые позволяют найти некоторый компромисс между разрешающей способностью алгоритма и его чувствительностью к шумам. Например, если требуется выделить в сигнале частоты, имеющие наибольшее соотношение сигнал/шум, то весовую функцию $\psi_G(k)$ следует выбрать таким образом, чтобы она зависела от спектра шума и полезного сигнала. Такая функция может быть построена заранее с учетом априорных знаний или вычислена в процессе обработки сигнала, что обеспечивает адаптивность метода [7]. Примеры весовых функций, которые чаще всего применяются при оценке задержки, представлены в табл. 1. Более сложные функции используют статистические методы оценки соотношения сигнал/шум и требуют значительных вычислительных ресурсов, что пока сдерживает их применение в задачах управления в реальном времени [11].

Таблица 1

Функция	Формула
Импульсная характеристика Рофа (Roth impulse response)	$1/G_{x_1x_2}(k)$
Сглаженная функция когерентности (Smoothed Coherence Transform — SCOT)	$1\Big/\sqrt{G_{x_1x_1}(k)G_{x_2x_2}(k)}$
Преобразование фазы (Phase Transform — PHAT)	$1\Big/\left\lvert G_{x_1x_2}(k)\right\rvert$

Среди непараметрических методов определения положения источника звука можно выделить алгоритм адаптивной декомпозиции собственных комплексных чисел (Adaptive Eigenvalue Decomposition — AED) [10]. Оценка задержки вычисляется путем анализа импульсных характеристик между источником и микрофонами. Точное вычисление собственных векторов не является тривиальным вследствие нестационарности речи, фонового шума и неизвестной длины импульсной характеристики, поэтому на практике применяются упрощенные алгоритмы, основанные на итеративном поиске максимальных или минимальных значений собственных чисел. Наиболее простой алгоритм основан на оценке среднеквадратической ошибки по методу LMS [10].

При проведении экспериментов для локализации пользователя, взаимодействующего с информационной системой — интеллектуальным многомодальным киоском, были применены три алгоритма: GCC-SCOT (Smoothed Coherence Transform — сглаженная функция когерентности), GCC-PHAT (Phase Transform — метод преобразования фазы) и LMS (Lest Mean Squares — метод наименьшей среднеквадратической ошибки). Далее приведены результаты экспериментов по оценке направления источника звука с использованием разработанного авторами массива микрофонов [12].

Эксперименты. Для дистанционного распознавания речи в разработанном киоске использовался массив микрофонов, спроектированный с учетом обеспечения эргономичных условий взаимодействия и минимизации влияния работы динамиков на микрофоны. В состав аппаратной части массива входят четыре микрофона „Октава МК-012“ и звуковая плата „PreSonus Firepod“. Для задачи локализации источника звука использовались два микрофона, расположенные горизонтально на одной линии.

При проведении экспериментов изменялись следующие параметры: 1) расстояние между микрофонами; 2) расстояние от источника звука до микрофона; 3) отклонение источника звука от линии массива микрофонов. Чтобы оценить, насколько точно работает каждый из

трех методов (GCC-SCOT, GCC-PHAT и LMS), в качестве источника звука использовался один динамик, через который проигрывалась предварительно записанная фраза, произнесенная мужским голосом. Источник речевого сигнала последовательно находился в 33 положениях согласно схеме, приведенной на рис. 1.

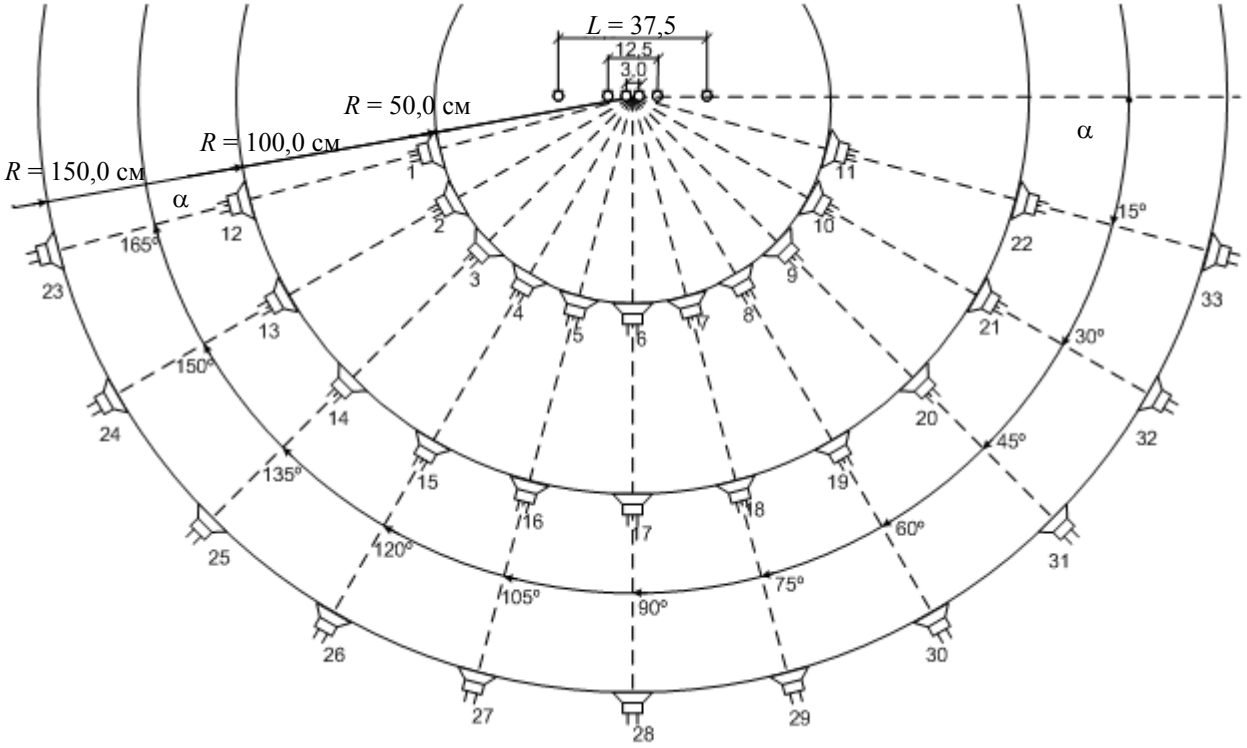


Рис. 1. Схема положений источника речи относительно массива микрофонов

Во всех экспериментах сигнал записывался синхронно двумя микрофонами с частотой дискретизации 16 кГц. Комплексное преобразование Фурье вычислялось для сегмента сигнала размером 512 отсчетов с шагом 128 отсчетов. Оценка угла α производилась только для сегментов, максимальное значение функции взаимной корреляции которых превышало заданный порог. Длительность тестовой фразы составляла 2,3 с. Усредненная оценка угла вычислялась для сегментов записанного сигнала в диапазоне 0,5—1,8 с.

В табл. 2 приводится сравнение методов локализации в зависимости от расстояния (L) между микрофонами. Следует отметить, что с увеличением расстояния ошибка локализации, обозначенная как Δ и представляющая собой разность между реальным углом α и углом β, вычисленным с помощью массива микрофонов, уменьшается при использовании каждого из методов. При переходе от L = 3 см к L = 12,5 см ошибка уменьшилась более чем в два раза. Однако при L = 37,5 см ошибка снизилась менее чем на 1º, поэтому дальнейшее увеличение расстояния не проводилось. Для рассматриваемой задачи ошибка в 5º является вполне приемлемой, так как при взаимодействии пользователя с киоском рабочий сектор составляет более 20º. Кроме того, дальнейшее увеличение расстояния между микрофонами нецелесообразно, поскольку это приведет к увеличению размеров киоска.

Таблица 2

L, см	Δ, …°		
	LMS	SCOT	PHAT
3,0	14,12	17,91	14,91
12,5	6,05	6,06	5,92
37,5	5,85	5,78	5,81

Затем было проанализировано, как ошибка локализации зависит от степени отклонения источника звука от нормали (90°) массива микрофонов. Как видно из рис. 2, с увеличением

отклонения от нормали ошибка локализации возрастает. Метод LMS „работает“ более точно, но несколько проигрывает при малых отклонениях двум другим методам. На рисунке показаны абсолютные значения ошибки, на самом же деле при $\alpha = 15\ldots75^\circ$ ошибка стабильно принимала положительные значения, а при $\alpha = 105\ldots165^\circ$ — отрицательные. Это связано с тем, что во всех использованных методах принимается следующее допущение: от источника звука к микрофонам приходит плоская волна, а не сферическая. С увеличением расстояния от источника до микрофонов данная погрешность снижается, и, как показано на рис. 3, ошибка локализации источника звука, находящегося на расстоянии 150 см от массива при $\alpha = 30\ldots150^\circ$, не превышает 5º.

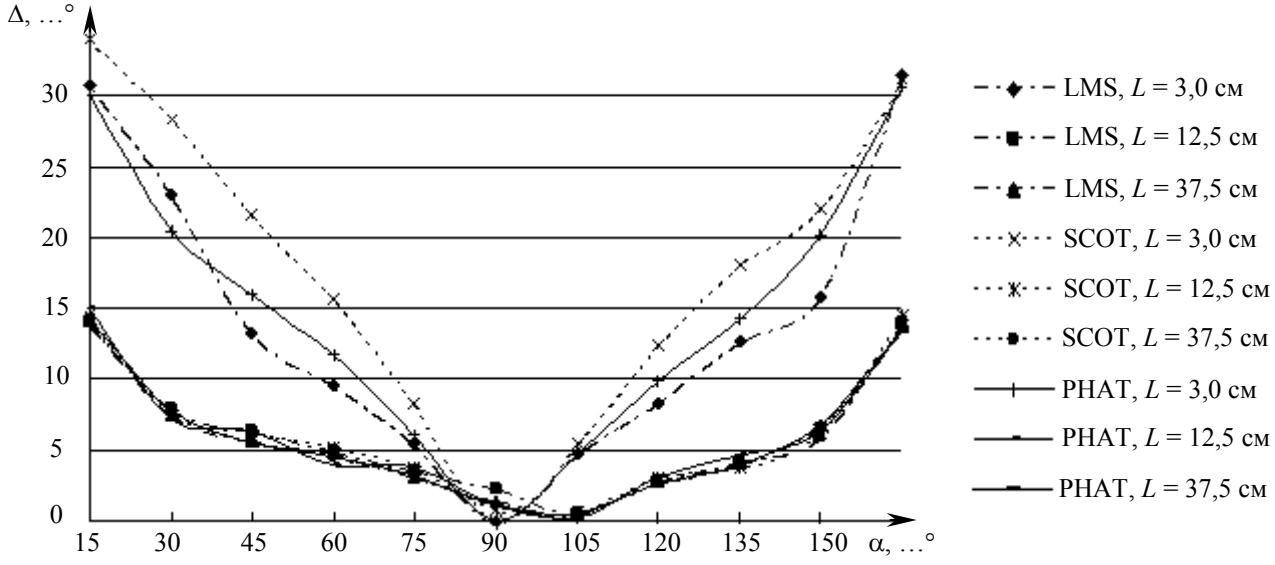


Рис. 2. Зависимость ошибки локализации от угла α и расстояния L между микрофонами

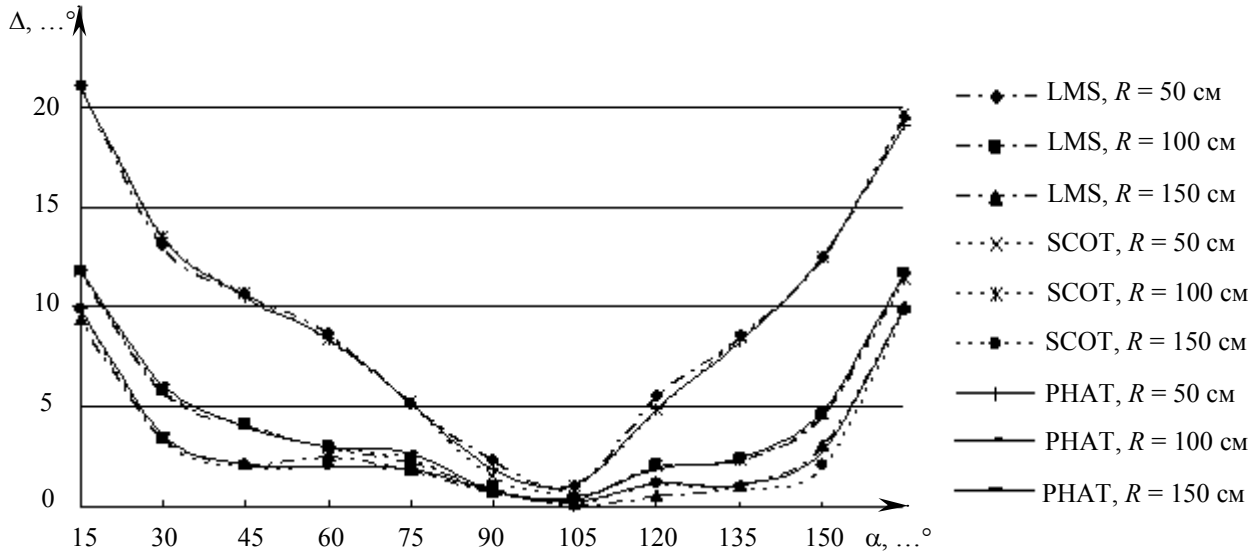


Рис. 3. Зависимость ошибки локализации от угла α и расстояния R до источника звука

Необходимо также учесть влияние инструментальных ошибок на постановку и проведение эксперимента. В частности, возможна некоторая погрешность, возникшая при установке массива микрофонов и динамика, что могло повлиять на точность работы методов. Например, для $L = 3$ см (см. рис. 2) минимальная ошибка наблюдается при $\alpha = 90^\circ$, а для $L = 12{,}5$ см и $L = 37{,}5$ см минимум ошибки смещен в сторону 105º. Данное несовпадение минимумов функций можно объяснить смещением микрофонов относительно центра массива. Тем не менее

точность протестированных методов в целом является удовлетворительной для задачи локализации пользователя при взаимодействии с многомодальным киоском.

Заключение. К настоящему времени разработан прототип интеллектуального многомодального киоска, позволяющий в ходе натурных экспериментов создать наиболее удобный для пользователя способ общения с автоматической справочной системой. При организации взаимодействия пользователя с многомодальным киоском для начала диалога система, в первую очередь, должна определить присутствие пользователя в рабочей зоне перед киоском. Кроме того, эффективность автоматического распознавания речи существенно зависит от точности определения границ голосовой команды в записанном звуковом сигнале. Подавление сигналов, поступающих от источников, находящихся вне рабочей зоны киоска, значительно снижает вероятность появления неречевых сигналов на входе системы распознавания речи.

Рассмотренные в настоящей статье методы определения задержки прихода звуковой волны от источника изначально были разработаны в области радиолокации и адаптированы для задачи дистанционного распознавания речи. Методы LMS, GCC-SCOT и GCC-PHAT были применены авторами в разработанном массиве микрофонов, что позволило на практике оценить влияние расположения микрофонов на точность локализации, а также определить погрешность, связанную с допущением о распространении от источника плоской волны, а не сферической.

Анализ влияния уровня шума, направления звуковой волны шума, громкости речи и других факторов на работу алгоритмов предполагается произвести в последующих экспериментах. Кроме того, планируется совместить модули обработки видео- и аудиопотоков для создания бимодальной системы локализации пользователя, что позволит существенно понизить влияние шумов на определение его положения и повысить точность системы дистанционного распознавания речи в многомодальном интеллектуальном киоске.

Работа выполнена при поддержке Российского фонда фундаментальных исследований, грант № 07-07-00073-а „Исследование многомодального взаимодействия на базе информационного киоска“.

СПИСОК ЛИТЕРАТУРЫ

1. *Yiteng Huang, Benesty J.* Audio Signal Processing for Next-Generation Multimedia Communication Systems. Norwell, MA: Kluwer Academic Publishers, 2004.

2. *Macho D., Padrell J., Abad A.* et al. Automatic speech activity detection, source localization and speech recognition on the CHIL seminar corpus // Proc. of IEEE Intern. Conf. on Multimedia and Expo. Amsterdam, Netherlands. 2005. P. 876—879.

3. Microphone Arrays / Eds.: *M. Brandstein, D. Ward*. Berlin: Springer Verlag, 2001.

4. *Krim H., Viberg M.* Two decades of array signal processing research: The parametric approach // IEEE SP Magazine. 1996. Vol. 13, July. P. 67—94.

5. *Schmidt R.* Multiple emitter location and signal parameter estimation // IEEE Transact. on Antennas and Propagation. 1986. Vol. AP-34, March. P. 276—280.

6. *Roy R., Kailath K.* ESPRIT — estimation of signal parameters via rotational invariance techniques // IEEE Transact. on ASSP. 1989. Vol. 37, N 7. P. 984—995.

7. *Knapp C. H., Carter G. C.* The generalized correlation method for estimation of time delay // IEEE Trans. Acoustics Speech Signal Proc. 1979. Vol. 24. P. 320—327.

8. *Omologo M., Svaizer P.* Acoustic event localization using a crosspower-spectrum phase based technique // Proc. of ICASSP. Adelaide, Australia. 1994.

9. *Lathoud G., McCowan I. A.* A sector-based approach for localization of multiple speakers with microphone arrays // Proc. of SAPA-2004, Korea. 2004.

10. *Benesty J.* Adaptive eigenvalue decomposition algorithm for passive acoustic source localization // J. Acoust. Soc. Amer. 2000. Vol. 107. P. 384—391.

11. *Trifa V., Koene A., Moren J., Cheng G.* Real-time acoustic source localization in noisy environments for human-robot multimodal interaction // Proc. of RO-MAN 2007, Korea. 2007.

12. *Ронжин А. Л., Карпов А. А., Леонтьева Ан. Б., Костюченко Б. Е.* Разработка многомодального информационного киоска. // Тр. СПИИРАН. СПб.: Наука, 2007. Вып. 5, т. 1. С. 227—245.

Сведения об авторах

Андрей Леонидович Ронжин — СПИИРАН, лаборатория речевых и многомодальных интерфейсов; E-mail: ronzhin@iias.spb.su

Алексей Анатольевич Карпов — СПИИРАН, лаборатория речевых и многомодальных интерфейсов; E-mail: karpov@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 004.8

И. А. Кагиров, Ан. Б. Леонтьева

АВТОМАТИЧЕСКИЙ СИНТАКСИЧЕСКИЙ АНАЛИЗ РУССКИХ ТЕКСТОВ НА ОСНОВЕ ГРАММАТИКИ СОСТАВЛЯЮЩИХ

Представлены концепция и пути реализации программного модуля синтаксического анализа для литературного русского языка. Основным инструментом исследования является так называемая „грамматика непосредственных составляющих“, используемая для формального представления синтаксических структур. Сформулировано теоретическое обоснование процесса выделения множества синтаксических структур, существенных для формального представления текстов на русском языке.

Ключевые слова: *непосредственные составляющие, синтаксические структуры, автоматический анализ текста.*

Введение. Создание автоматического модуля синтаксического анализа (МСА) является одной из актуальных задач в компьютерной лингвистике, решение которой позволит достичь высокого уровня формализации языковых структур в различных прикладных задачах — от создания систем автоматического распознавания речи до поисковых систем в сети Интернет.

Однако создание МСА для русского языка связано с большими трудностями вследствие недостаточно разработанной теоретической базы в общем и прикладном языкознании; кроме того, структуры языка отличаются разнообразием и зачастую высоким уровнем сложности, предусмотреть который чрезвычайно трудно. В связи с этим в настоящей статье предлагается структура МСА, работающего с простыми синтаксическими структурами; создание такого модуля, способного обрабатывать тексты на русском языке любой сложности, представляется на настоящем этапе невозможным.

Теоретическая база: грамматика зависимостей и непосредственные составляющие. Под синтаксисом понимается такой уровень языка, наибольшими и основными единицами которого являются предложения, а наименьшими — грамматические слова (словоформы). Далее предложением называется грамматически связная цепочка слов, выражающая некоторое суждение. Грамматически связная цепочка — такая цепочка, в которой словоформы находятся в определенных грамматических отношениях между собой. В свою очередь, словоформа — это слово в одной из своих грамматических форм (характеризующейся определенными для каждого языка грамматическими признаками; так, для существительного в русском

языке это падеж и число). Таким образом, синтаксическая структура предложения представляет собой цепочку, состоящую из конечного множества словоформ, связанных синтаксическими отношениями.

С математической точки зрения, любое предложение может быть представлено как направленный граф. Главные вершины графа соединены подчинительными связями с зависимыми вершинами: если между вершинами (словоформами) существует отношение зависимости $X \to Y$, то следует говорить, что X подчиняет Y, а Y зависит от X, т.е. X называется вершиной, а Y — зависимой. Существуют три типа подчинительной связи между словоформами: *управление*, *согласование* и *примыкание* [1].

Множество синтаксических явлений в пределах предложения трудно описать, опираясь исключительно на взаимоотношения между терминальными элементами — минимальными синтаксическими единицами (словоформами) [1]. Поэтому в синтаксический анализ вовлекаются иерархически организованные единицы более высокого уровня — фразовые категории (ФК, англ. Phrasal Category), представляющие собой группу, в которой имеется одна главная вершина, а также может быть одна или несколько зависимых вершин. Фразовые категории имеют обычно прозрачную, жестко иерархизированную структуру, что позволяет описать синтаксис языка; ФК, выделенная в конкретном предложении и функционирующая как синтаксическая сущность, называется непосредственной составляющей.

Важнейшей единицей синтаксического уровня языка (в формальном синтаксисе) является так называемая клауза („элементарное предложение“, „предикация“) [2]. Под клаузой в настоящей статье понимается любая синтаксическая группа, распадающаяся на глагольную и именную группы.

Синтаксический анализ предложения. Поскольку число различных конструкций предложений бесконечно, при синтаксическом разборе имеет смысл ориентироваться на более мелкие единицы — фразовые категории, введенные ранее. Таким образом, алгоритм автоматического анализа сводится к вычленению ФК в составе предложения и поиску связей между ними.

Для разработки модуля автоматического синтаксического анализа использовался корпус текстов, состоящий из составленных в соответствии с нормами литературного языка клауз с нераспространенной синтаксической структурой [3]. Этот корпус на настоящем этапе разработки модуля синтаксического анализа отвечает следующему требованию: идентификация отдельных ФК в структуре клаузы и определение связей между ними.

На основе анализа используемого корпуса текстов были выделены пять синтаксических групп, представленных на рис. 1, где приняты следующие обозначения: ИГ — именная группа, в которой вершиной является имя существительное (Сущ) или местоимение (М); ГГ″ — глагольная группа, где вершина — финитный [4] глагол (Глаг); ГГ′ — глагольная группа, где вершиной является группа ГГ″; ПГ — группа прилагательного, вершина — краткое прилагательное (КрП) или прилагательное (П); ПрГ — предложная группа, вершина — предлог (Пр); ИнфГ — инфинитивная группа, вершина — инфинитив (Инф); ВспГ — вспомогательный глагол „быть“, Нар — наречие; кроме того, символ „*“ означает, что элементы группы могут стоять также и в обратном порядке; косая линия показывает положение группы относительно вершины: /ИГ=Вершина→ИГ, ИГ/=ИГ→Вершина; стрелкой обозначено направление зависимости (от вершины к зависимому); в фигурных скобках указана часть речи.

Для определения падежа, в котором стоит зависимое слово при подчинительной связи, используется словарь [5]. Со временем предполагается создать словарь, предназначенный непосредственно для автоматического синтаксического анализа.

<table>
<tr><th>ИГ
{Сущ}/{М}</th><th>ГГ′
{ГГ″}</th><th>ПГ
{КрП}</th><th>ПрГ
{Пр}</th><th>ГГ″
{Глаг}</th><th>ИнфГ
{Инф}</th></tr>
<tr><td>—</td><td>ИГ* управление</td><td>/ИГ управление</td><td rowspan="6">/ИГ управление</td><td>—</td><td rowspan="6">ИГ* управление</td></tr>
<tr><td>/{ИГ} управление</td><td>ИГ/ управление /ИГ управление</td><td rowspan="5">{Нар}/ примыкание</td><td>{ВспГ} → {КрП}</td></tr>
<tr><td>{П}/ согласование</td><td>ПрГ* примыкание</td><td>{Нар}* примыкание</td></tr>
<tr><td rowspan="3">/Союз-{Сущ}</td><td>ИГ/ управление /ПрГ примыкание</td><td rowspan="2">{Нар}/ примыкание /{Нар} примыкание</td></tr>
<tr><td>/ИГ управление /ПрГ примыкание</td></tr>
<tr><td>/ИнфГ примыкание</td></tr>
</table>

Рис. 1. Типы ФК, используемых в модуле синтаксического анализа

Программная реализация модуля синтаксического анализа. Структурная схема модуля приведена на рис. 2. Входные данные представляют собой список простых предложений, который поступает в блок синтаксического анализа. В блоке обработки предложения обрабатываются пословно. Исходная словоформа передается в блок морфологического анализа [6], в котором для нее подбираются все возможные варианты основ и соответствующие грамматические показатели. В зависимости от части речи и грамматических показателей словоформы выделяется соответствующая синтаксическая группа. Группа может определяться однозначно, либо могут существовать варианты ее определения. Например, если на вход поступило имя существительное, то первая группа в предложении будет именной. В этом случае запоминается порядковый номер группы и обрабатывается следующее слово. Если же первым словом в предложении является наречие, то оно может относиться как к глагольной группе, так и к группе прилагательного. В этом случае для однозначного определения группы требуется проанализировать следующее слово.

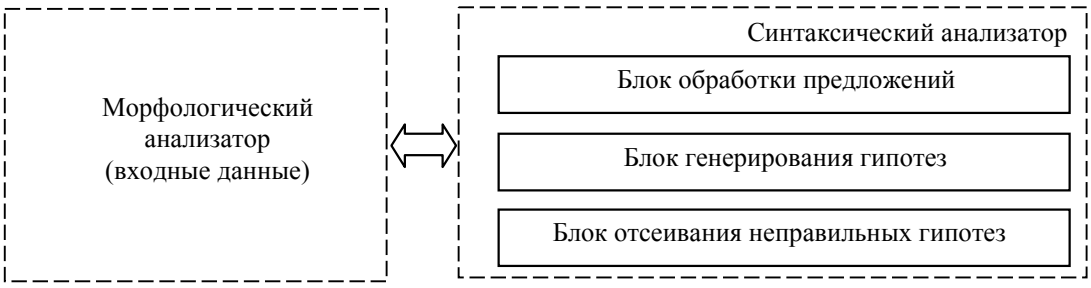


Рис. 2. Структурная схема модуля синтаксического анализа

Начиная со второго слова в предложении важную роль играет не только часть речи, которой выражена данная словоформа, но и информация о группе или группах, выделенных на данный момент. Поступившая на вход словоформа может принадлежать текущей синтаксической группе или выделяться в другую группу. В этом случае формируется дополнительная гипотеза и рассматриваются оба варианта. В конечном счете в предложении выделяются группа подлежащего и группа сказуемого. Важно отметить, что некоторые группы, например глагольная, могут содержать в себе другие группы.

Выходной файл представляет собой список предложений, каждое из которых разбито на синтаксические группы. Если предложение содержит слово, отсутствующее в словаре системы, то оно выводится без разбора.

Тестирование модуля МСА осуществлялось с использованием ГОСТ Р 50840-95 [7]. При анализе результатов были выявлены ошибки, представленные в табл. 1.

Таблица 1

Ошибка	Количество ошибок, %
Наличие в анализируемом тексте синтаксических конструкций, отличных от исходных синтаксических групп	26
Наличие в тексте слов, отсутствующих в словаре	6
Общее количество неправильно разобранных предложений	32
Семантическая и морфологическая неоднозначность слов, порождающая несколько вариантов разбора предложения	22

Семантическая и морфологическая неоднозначность слов приводит к построению большого количества гипотез, и не все из них могут быть отсеяны за счет проверки синтаксических связей в предложении. В этом случае, кроме правильно разобранного предложения, выводится еще и неправильный вариант разбора, который не был отсеян программно. В тестовом корпусе таких предложений оказалось 22 %. В обработанном тексте только для 12 % предложений было построено по одной гипотезе, остальные 88 % предложений порождали несколько гипотез разбора; из них в 66 % случаев удалось автоматически избавиться от неправильных вариантов за счет анализа морфологических показателей словоформ и синтаксических связей между словами.

Анализ причин возникновения ошибки такого типа был проведен по 50 тестовым предложениям. Статистика по количеству слов, порождающих несколько гипотез вследствие лексической и морфологической неоднозначности, представлена в табл. 2. Частеречная неоднозначность возникает в основном среди наречий, частиц и кратких форм прилагательных. Кроме того, существительным присуща семантическая и просодическая неоднозначность, а также омонимия.

Таблица 2

Причина порождения гипотез	Количество слов
Слова, относящиеся к нескольким частям речи	48
Слова, имеющие одинаковые словоформы в именительном и винительном или родительном падежах	24
Слова, содержащие в себе обе причины порождения гипотез	10
Общее количество слов в тестируемом тексте	206

Заключение. Разработка систем автоматического распознавания речи, содержащих большой словарь, а также систем стенографирования требует формирования грамматически правильных предложений в процессе обработки. Построение статистической модели языка и согласование окончаний словоформ в распознанной фразе можно осуществить путем проведения автоматического синтаксического анализа. При анализе результатов тестирования разработанного модуля, построенного на основе концепции выделения фразовых категорий, были выявлены ошибки, для устранения которых необходимо расширить базу синтаксических групп за счет анализа большего количества текстов. Также планируется разработка алгоритма автоматического расширения словаря.

Работа выполнена в рамках проекта Российского фонда фундаментальных исследований (№ 08-08-00128) и проекта ОИТВС РАН (№ 4.2).

СПИСОК ЛИТЕРАТУРЫ

1. *Тестелец Я. Г.* Введение в общий синтаксис. М.: Изд-во Рос. гос. гуманит. ун-та, 2001.

2. Фундаментальные направления современной американской лингвистики: сб. обзоров / Под ред. *А. А. Кибрика* и др. М.: УРСС, 1997.

3. ГОСТ 16600 — 72. Передача речи по трактам радиотелефонной связи: Требования к разборчивости речи и методы артикуляционных изменений. М.: Изд-во стандартов, 1973.

4. Лингвистический энциклопедический словарь / Под ред. *В. Н. Ярцевой* и др. М.: Сов. энциклопедия, 1990.

5. Большой толковый словарь русского языка / Под ред. *Д. Н. Ушакова*. М.: Альта-принт, 2005.

6. *Kagirov I. A., Leontyeva An. B.* Grammar-based speech- and word-splitting // Proc. of 3rd Language & Technology Conf., Oct. 5—7, 2007, Poznań, Poland. Poznań: Fundacja Uniwersytetu im. A. Mickiewicza, 2007. P. 413—417.

7. ГОСТ Р50840 — 95. Передача речи по трактам связи. Методы оценки качества, разборчивости и узнаваемости. М.: Изд-во стандартов, 1995.

Сведения об авторах

Ильдар Амирович Кагиров — СПИИРАН, лаборатория речевых и многомодальных интерфейсов; E-mail: kagirov@iias.spb.su

Анастасия Борисовна Леонтьева — СПИИРАН, лаборатория речевых и многомодальных интерфейсов; E-mail: an_leo@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 004.522

Ал. Б. Леонтьева, И. С. Кипяткова

УЧЕТ ОСОБЕННОСТЕЙ СПОНТАННОЙ РЕЧИ ПРИ СОЗДАНИИ СИСТЕМ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ

Рассматривается подход к автоматической обработке спонтанной русской речи, заключающийся в распознавании нефонемных элементов и моделировании альтернативных вариантов произношения слов. Представлен ряд акустических и лексических моделей, предназначенных для отделения помех от ключевых слов и учитывающих возможные элементы спонтанной речи. Приведен алгоритм создания альтернативных транскрипций с помощью расширенных правил транскрибирования. Представлены результаты экспериментов.

Ключевые слова: *распознавание речи, альтернативная транскрипция, нефонемные элементы.*

Введение. Большинство современных систем автоматического распознавания речи способны обрабатывать только прочитанные фразы или изолированно произнесенные слова. Распознавание спонтанной речи затрудняется наличием эмоций, различного рода шумов, невербальных и вокализованных пауз, артефактов [1]. Присутствие таких элементов является полезной информацией в задачах идентификации или верификации пользователей, но в задаче распознавания речи это ведет к снижению точности. Кроме того, при спонтанном речевом взаимодействии человеку свойственно использовать бóльшее количество слов, чем необходимо для четкого выполнения конкретной задачи. Междометия и вводные слова, выполняющие определенную дискурсивную роль при общении людей, для диалоговой системы, настроенной на решение узкой задачи, не будут нести информативной нагрузки.

В спонтанной речи произношение слов сильно варьируется различными людьми, а также зависит от контекста. В результате этого транскрипции произнесенных слов часто не совпадают с транскрипциями, созданными по правилам фонетики русского языка. Перечисленные явления не препятствуют общению между людьми, но могут стать критичными для автоматической системы распознавания речи.

Анализ составляющих спонтанной речи. Запись спонтанной речи содержит звуки фонемной и нефонемной природы, причем как звуки, производимые пользователем, непосредственно

контактирующим с диалоговой системой, так и посторонние шумы и звуки, произносимые людьми, не обращающимися к системе непосредственно. Сигнал на входе системы распознавания, помимо слов, может содержать акустический фон, сопровождающий запись сигнала; „шумы“ органов голосового аппарата и короткие звуковые явления (цоканье языком, причмокивание) — так называемые артефакты речи; невербальные паузы, вызванные, например, кашлем, смехом и т.п.; вокализованные паузы, заполненные элементами „э-э-э“, „м-м-м“ и др.; нефонемные подтверждения и отрицания („ага“, „угу“, „неа“) [2], а также эмоциональные междометия („ах“, „ой“ и др.) и вводные слова. Также возможно появление незнакомых слов вследствие спонтанности формирования фразы пользователем и ограниченности размера словаря системы распознавания. Разделение записанного речевого сигнала на определенные составляющие элементы позволяет более точно анализировать входные данные, сокращая вероятность ошибок наложения или замещения. При таком подходе в записанном сигнале следует различать звуки фонемной и нефонемной природы.

Диалоговая система, ориентированная на работу со спонтанной речью с учетом всех ее особенностей, позволит обеспечить в конечном счете бóльшую гибкость диалога, допуская формирование запросов и ответов пользователя в относительно произвольной форме. Усложнение системы обработки входного сигнала в целях исключения неинформативной составляющей (избыточной информации с точки зрения конкретной задачи) позволит упростить диалоговую модель.

Метод генерации альтернативных транскрипций слов. Помимо зашумленности речи, связанной с наличием посторонних слов, множества акустических эффектов и шумов, вариативность произношения слов в спонтанной речи значительно осложняет процесс распознавания. Несоответствие между наблюдаемым произношением и принятыми фонетическими транскрипциями является одной из главных причин низкой производительности систем распознавания спонтанной речи [3, 4].

На основе результатов исследований, полученных в области экспериментальной фонетики [5], был сформулирован ряд правил, которые достаточно точно описывают возможные отклонения в фонетических транскрипциях, связанные с ассимиляцией и редукцией звуков [6]. Эти правила используются для задачи синтеза речи и в настоящей статье адаптированы для автоматического распознавания спонтанной речи. На рис. 1 показан алгоритм транскрибирования словоформ с учетом правил ассимиляции и редукции [7].

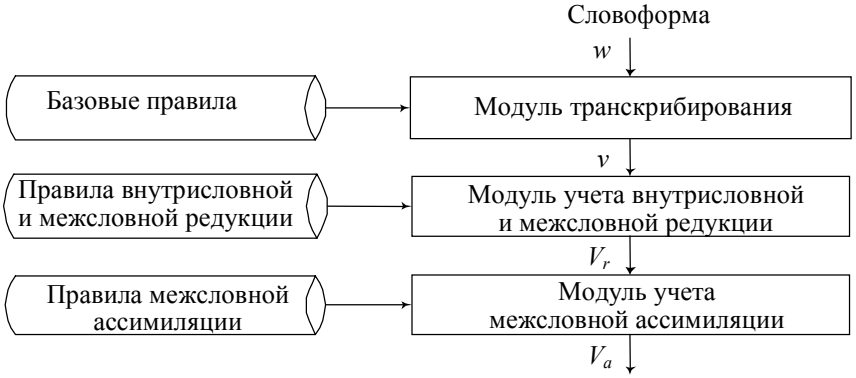


Рис. 1. Алгоритм транскрибирования словоформы с учетом ассимиляции и редукции

Модуль транскрибирования с учетом базовых правил фонетики преобразует поступающую на его вход словоформу w в последовательность фонем. Полученная таким образом транскрипция v поступает в модуль учета внутрисловной и межсловной редукции, где определяется, какие фонемы подвержены редукции. Если таких фонем в слове больше одной, то производится генерация всех возможных сочетаний редуцирующихся фонем. Затем каждое из полученных сочетаний обрабатывается отдельно. В результате одна альтернативная транскрипция генерируется путем удаления из базовой транскрипции фонем, указанных в текущем сочетании.

Таким образом, на выходе модуля получается набор альтернативных транскрипций V_r данной словоформы, учитывающий все возможные сочетания редуцирующихся фонем.

Далее, в модуле учета межсловной ассимиляции производится анализ первых и последних фонем в транскрипции. При обнаружении фонем, подверженных ассимиляции, производится генерация всех возможных контекстно-зависимых вариантов транскрипций. Полученный таким образом набор транскрипций V_a теоретически должен содержать все варианты произношений, которые могут возникать в разговорной речи.

Результаты оценки метода генерации. Для оценки метода генерации альтернативных транскрипций по расширенным (относительно базовых) правилам транскрибирования использовался словарь, построенный по названиям рубрик электронного каталога „Желтые страницы Санкт-Петербурга“. Размер базового словаря составил 17 662 транскрипции словоформ, размер расширенного словаря — 192 303 транскрипции. Соотношение между базовыми и альтернативными транскрипциями показано на рис. 2. Для большинства словоформ генерировалось 3 альтернативные транскрипции, однако имелись словоформы, для которых создавалось более 1000 транскрипций.

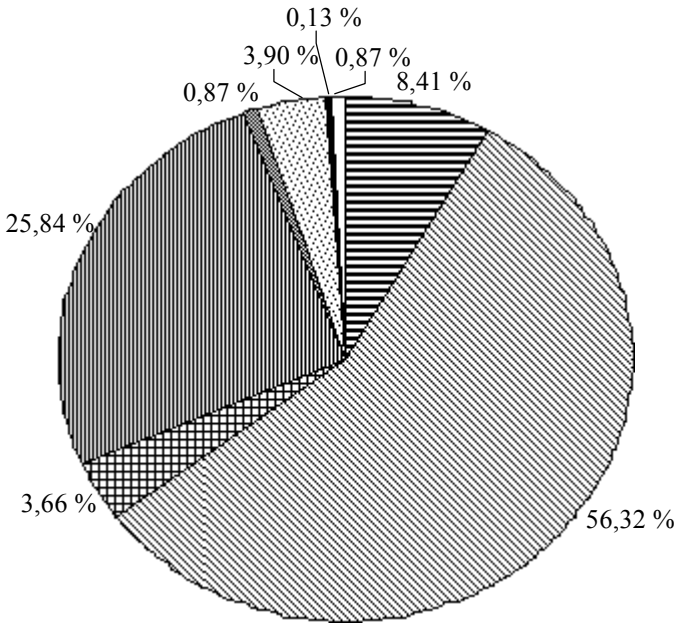


Рис. 2. Соотношение типов альтернативных транскрипций по принципу образования:
транскрипции: ▬ — базовые, ▧ — с внутрисловной редукцией, ⊞ — с межсловной редукцией, ▥ — с внутрисловной и межсловной редукцией, ▩ — с межсловной ассимиляцией, ⚄ — с внутрисловной редукцией и межсловной ассимиляцией, ■ — с межсловной редукцией и межсловной ассимиляцией, □ — с внутрисловной и межсловной редукцией и ассимиляцией

Для того чтобы определить, какая часть словоформы наиболее часто подвергается изменению, для тестового словаря была вычислена средняя частота возникновения редукций и ассимиляций в основах R_{stem} и окончаниях R_{end} словоформ с использованием следующих выражений:

$$R_{\text{stem}} = \frac{1}{L}\sum_{i=1}^{L}\frac{s_{\text{stem}_i}}{p_{\text{stem}_i}};\quad R_{\text{end}} = \frac{1}{K}\sum_{i=1}^{K}\frac{s_{\text{end}_i}}{p_{\text{end}_i}},$$

где s_{stem_i} — число редукций и ассимиляций в основе i-й словоформы; p_{stem_i} — число фонем в основе i-й словоформы; s_{end_i} — число редукций и ассимиляций в окончании i-й словоформы; p_{end_i} — число фонем в окончании i-й словоформы; L — число уникальных основ (4790 для данного словаря); K — число уникальных окончаний (225 для данного словаря).

В результате расчетов $R_{\text{stem}} = 0,213$, $R_{\text{end}} = 0,296$. Следовательно, редукции и ассимиляции чаще возникают в окончании словоформ, чем в основе.

С помощью разработанной программы был проведен также анализ частоты применения каждого из правил транскрибирования.

Результаты экспериментов с использованием модели „речевого мусора“. Для отделения помех от ключевых слов был создан ряд акустических и лексических моделей, учитывающих возможные элементы спонтанной речи, содержащиеся в записанном входном сигнале. Эти модели были встроены в модифицированный гибридный декодер, состоящий из словаря, фонемного декодера, модели „речевого мусора“ (модель РМ) и модели тишины [8]. Словарь декодера, построенный на базе двухуровневого морфофонемного префиксного графа [9], обрабатывает все слова, поступающие на вход системы, в том числе ключевые слова, междометия и вводные слова. Фонемный декодер обеспечивает распознавание слов, не содержащихся в словаре системы распознавания. Модель РМ осуществляет обработку фонового шума, вокализованных и невербальных пауз и артефактов. Модель тишины обеспечивает распознавание беззвучных пауз между словами. В результате во входном сигнале одновременно производится поиск и распознавание всех слов из словаря, а также возможных речевых и неречевых помех.

Для устранения маловероятных последовательностей слов обычно применяется статистическая модель языка, которая содержит вероятности всех комбинаций слов из словаря (пар слов, троек слов и т.д., в зависимости от сложности модели). В гибридном декодере модель языка также должна учитывать возможное появление речевого мусора в последовательности слов.

В таблице представлены результаты работы системы распознавания речи при различных настройках модели языка и использовании модели „речевого мусора“ и без нее. Тестирование проводилось на собранном корпусе, половина фраз в котором содержала различные речевые помехи, остальные фразы состояли из словарных слов. Таким образом, моделировалась ситуация спонтанного общения, когда человек может в любом месте фразы запнуться, вызвав тем самым появление, например, вокализованной паузы, или же, наоборот, четко проговорить ожидаемый от него запрос. Априори можно предположить, что система, учитывающая возможное появление помех в речевом сигнале, будет лучше распознавать входной сигнал по сравнению с системой, не учитывающей речевые помехи. Для исследования этого положения было проведено тестирование системы в двух режимах: с использованием модели РМ и без нее.

Режим тестирования	Модель языка		
	отсутствует	определяющая допустимые фразы	статистическая, допускающая появление помех в любом месте фразы
Модель РМ используется	39,04 %	70,32 %	70,52 %
Модель РМ не используется	27,89 %	42,83 %	45,02 %

Модель языка накладывает ограничения на вольность построения фраз. В первом эксперименте модель языка фактически была отключена, допуская все возможные варианты построения фразы. Во втором эксперименте модель языка задавалась с помощью списка всех возможных структур фраз. При этом допускалось присутствие наиболее вероятных помех в определенных местах фразы. В третьем эксперименте использовалась статистическая модель языка, допускающая появление помех в любом месте фразы.

При включенной модели РМ осуществлялось распознавание помех, при этом точность зависела от соответствия модели языка тестовым фразам. При отключенной модели РМ в первом эксперименте ошибки возникали из-за распознавания словарных слов на участках с речевыми помехами. Во втором и третьем экспериментах модель языка, более подробно описывая возможную структуру входного сигнала с учетом наличия в нем неизвестных системе (немоделируемых) элементов, не позволяла идентифицировать помехи, но иногда позволяла избежать наложения ключевых слов на участок с помехами.

Как видно из таблицы, наиболее высокий процент распознавания получается при использовании модели РМ и допущении появления помех в любом месте. В настоящее время ведутся работы по накоплению базы данных речевых помех и поиску закономерностей возникновения таких помех в спонтанной речи. Это позволит обеспечить более точное моделирование спонтанной речи, а также устранение речевого мусора из последующей обработки сигнала.

Заключение. Представленные результаты исследований подтверждают целесообразность использования моделей помех и нефонемных элементов при обработке разговорной речи. Генерация альтернативных транскрипций при создании словаря системы автоматического распознавания позволяет учесть вариативность произношения слов. Однако применение всех правил редукции и ассимиляции приводит к значительному расширению словаря и созданию неправдоподобных транскрипций. Для „отсеивания“ редких вариантов произношения необходимо определить вероятность появления альтернативных транскрипций. Это, в частности, и будет предметом дальнейших исследований.

В перспективе создание систем автоматического распознавания речи, учитывающих специфику спонтанной речи, позволит снять ряд ограничений, накладываемых на диалог с пользователем, что в итоге сделает человекомашинное взаимодействие более естественным и продуктивным.

Исследования, описанные в настоящей статье, проводятся при поддержке Российского фонда фундаментальных исследований, проект № 08-08-00128 „Моделирование нефонемных речевых элементов и создание альтернативных транскрипций для распознавания спонтанной русской речи“.

СПИСОК ЛИТЕРАТУРЫ

1. *Butzberger J., Murveit H., Shriberg E., Price P.* Spontaneous speech effects in large vocabulary speech recognition applications // Proc. of the Workshop on Speech and Natural Language of Human Language Technology Conf., Morristown, NJ, USA. 1992. P. 339—343.

2. *Леонтьева Ал. Б.* Разработка моделей мусора для устранения помех при распознавании спонтанной речи // Искусственный интеллект. 2007. № 3. С. 309—318.

3. *Greenberg S., Hollenback J., Ellis D.* Insights into spoken language gleaned from phonetic transcription of the switchboard corpus // Proc. Intern. Conf. on Spoken Language Processing, Philadelphia, USA. 1996. P. 24—27.

4. *McAllaster D., Gillick L., Scattone F., Newman M.* Fabricating conversational speech data with acoustic models: a programm to examine model-data mismatch // Proc. Intern. Conf. on Spoken Language Processing, Sydney, Australia. 1998. P. 1847—1850.

5. Русская разговорная речь / Под ред. *Е. А. Земской.* М.: Наука, 1973.

6. *Лобанов Б. М., Цирульник Л. И.* Моделирование внутрисловных и межсловных фонетико-акустических явлений полного и разговорного стилей в системе синтеза речи по тексту // Тр. Первого междисциплинарного семинара „Анализ разговорной русской речи“ (АР^3 - 2007). СПб.: Изд-во ГУАП, 2007. С. 57—71.

7. *Леонтьева Ал. Б., Кипяткова И. С.* Моделирование нефонемных речевых элементов и создание альтернативных транскрипций для распознавания спонтанной речи // Там же. С. 77—85.

8. *Bazzi I., Glass J.* Modeling out-of-vocabulary words for robust speech recognition // Proc. 6th Intern. Conf. on Spoken Language. Beijing, 2000.

9. *Ронжин А. Л., Леонтьева Ан. Б., Кагиров И. А., Леонтьева Ал. Б.* Двухуровневый морфофонемный префиксный граф для декодирования русской слитной речи // Тр. СПИИРАН. СПб.: Наука, 2007. Вып. 4, т. 1. С. 388—404.

Сведения об авторах

Александра Борисовна Леонтьева — СПИИРАН, лаборатория речевых и многомодальных интерфейсов; E-mail: leonty@iias.spb.su

Ирина Сергеевна Кипяткова — СПИИРАН, лаборатория речевых и многомодальных интерфейсов; E-mail: kipyatkova@iias.spb.su

Поступила в редакцию
06.05.08 г.

КОМПЛЕКСНОЕ МОДЕЛИРОВАНИЕ И АНАЛИЗ СЛОЖНЫХ ДИНАМИЧЕСКИХ ОБЪЕКТОВ

УДК 681.5

Н. П. Кириллов

ВЫБОР МОДЕЛИ ФУНКЦИОНИРОВАНИЯ ТЕХНИЧЕСКОЙ СИСТЕМЫ ИЗ МНОЖЕСТВА ЕЕ АЛЬТЕРНАТИВНЫХ МОДЕЛЬНЫХ ПРЕДСТАВЛЕНИЙ

Обосновывается теоретический подход к выбору безызбыточной модели правил функционирования технической системы из множества альтернативных моделей, которые могут быть построены по исходной информации о таких системах. Предлагаемый подход основан на формализации понятий „цель управления“ и „целевое состояние“, определении правила их однозначной идентификации и особенностей структуры моделей системы.

Ключевые слова: *модель технической системы, состояние системы, цели управления.*

Информация о правилах функционирования любой технической системы (ТС) содержится в ее исходных описаниях (конструкторской документации, технических описаниях, экспертных знаниях и т.п.). В работе [1] было показано, что эта информация гипотетически может быть представлена в виде модели дискретной динамической системы с пространством состояний C (модели „белого ящика“ [2]). Такую модель можно рассматривать как гомоморфный прообраз множества детерминированных моделей ТС, каждая из которых (для некоторых фиксированных условий — подмножества возмущающих воздействий; интервала времени существования системы; времени, требуемого на реализацию процессов передачи и обработки телеметрической информации) взаимно-однозначно соответствует специальным образом заданному на множестве C отношению эквивалентности R ($R \subset \Re$), где $\Re$ — множество таких отношений. Иными словами, эти модели представляют собой результат агрегирования информации модели „белого ящика“, которая, по определению, содержит максимально доступный для анализа объем информации о ТС.

Возможность представления правил функционирования ТС множеством различных детерминированных моделей (для различных комбинаций и значений перечисленных выше фиксированных условий) обусловливает необходимость решения задачи выбора такой ее модели, которая в наибольшей степени удовлетворяет потребностям системы управления ТС. В идеальном случае такая модель должна содержать только необходимую и достаточную информацию, используемую при управлении системой. Это означает, что степень агрегации информации о состояниях ТС, при интерпретации ее поведения в виде модели „белого ящика“ (а следовательно, и степень агрегации информации о поведении системы в целом), должна быть каким-то определенным образом согласована с целями управления (ЦУ) системой. Таким образом, для решения этой задачи необходимо использовать информацию о ЦУ ТС, которая в том или ином виде всегда присутствует в ее исходных описаниях. Для этого, в свою

очередь, необходимо сначала формализовать понятия „цели управления ТС“, „целевые состояния ТС“ и определить правила однозначной идентификации ЦУ по состояниям системы. Знание этих правил позволит определить условия максимально допустимой степени агрегации информации о состояниях модели „белого ящика“, при которой сохраняется возможность однозначной идентификации ЦУ ТС.

Технические системы предназначены для выполнения строго определенных целевых задач, которые можно сопоставить с целями управления. В свою очередь, каждой ЦУ должно соответствовать одно или несколько состояний ТС. Такие состояния ТС считаются одноцелевыми, т.е. эквивалентными (не различимыми) по отношению к соответствующей им цели управления. Естественно предположить, что смысловое содержание ЦУ должно интерпретироваться через смысловое содержание целевых состояний системы. Это обстоятельство позволяет определить некоторое формальное соответствие (γ) между конечным множеством ЦУ (W) и множеством состояний ТС (C): $\gamma : W \to C$. При этом множество $\gamma(x) \subset C$ интерпретируется как множество одноцелевых состояний, соответствующих цели $x(x \in W)$.

Отметим, что в общем случае не все состояния ТС могут быть проинтерпретированы как целевые. Однако для упрощения последующих рассуждений, без потери их общности, будем считать, что соответствие γ принимает свои значения на всем множестве C.

В общем случае соответствие γ может быть произвольным. Это его свойство можно характеризовать как возможность существования в множестве W различных элементов x и y, образы которых на множестве C имеют непустое пересечение: $\gamma(x) \bigcap \gamma(y) \neq \varnothing$. Это означает, что C содержит непустое подмножество состояний $\gamma(x) \bigcap \gamma(y)$, каждое из которых может рассматриваться как целевое, соответствующее и цели x, и цели y. Такие состояния будем называть многоцелевыми.

Соответствие γ можно рассматривать как формализованную модель алгоритма проверки факта соответствия состояний ТС (например, состояния c) заданной цели управления (например, x). Этот алгоритм заключается в следующем: если $c \in \gamma(x)$, это означает, что ТС находится в заданном целевом состоянии, в противном случае делается вывод о том, что состояние c заданной цели x не соответствует.

В моделях ТС, соответствующих элементам из множества $\Re$, состояния системы ассоциируются с классами разбиения (непересекающимися подмножествами) множества C. Идентификация состояний таких моделей по значениям наблюдаемых параметров функционирования ТС осуществляется только с точностью до этих классов, а не с точностью до состояний модели „белого ящика“ — элементов множества C. При этом остается открытым вопрос: при выполнении каких условий возможна однозначная идентификация целей управления ТС по состояниям таких моделей?

Учитывая, что множеству моделей ТС взаимно-однозначно соответствуют отношения эквивалентности на множестве C из множества $\Re$, эту задачу можно сформулировать следующим образом: **известно** соответствие $\gamma : W \to C$; **требуется** определить условия, при выполнении которых возможна однозначная идентификация целей управления ТС, представленной в виде модели, соответствующей отношению эквивалентности $R \in \Re$ на множестве C.

Решение этой задачи заключается в определении правил построения такого соответствия $\phi_R : W \to D_R$, которое удовлетворяет условию

$$\forall d \in D_R, \exists x \in W \; d \in \phi_R(x) \leftrightarrow \varphi_R^{-1}(d) \subseteq \gamma(x), \qquad (1)$$

где D_R — множество состояний модели ТС; φ_R — естественное отображение $\varphi_R : C \to D_R$, соответствующее отношению эквивалентности R.

Для решения этой задачи введем ряд определений и используем некоторые теоремы, доказательство которых приведено в работе [3].

О п р е д е л е н и е 1. Множество C с заданным на нем отношением толерантности τ называется пространством толерантности, которое обозначается парой $\langle C, \tau \rangle$.

Зададим на множестве C отношение τ в соответствии со следующим правилом: $x \tau y \leftrightarrow \gamma^{-1}(x) \bigcap \gamma^{-1}(y) \neq \varnothing$, где x и y — элементы множества C. Очевидно, что отношение τ симметрично и рефлексивно, т.е. является отношением толерантности.

О п р е д е л е н и е 2. Множество $L \subseteq C$ называется предклассом в пространстве толерантности $\langle C, \tau \rangle$, если любые два его элемента x и y толерантны, т.е. для них выполнимо соотношение $x \tau y$.

О п р е д е л е н и е 3. Множество $K \subseteq C$ называется классом толерантности в пространстве $\langle C, \tau \rangle$, если K — максимальный предкласс по отношению включения.

О п р е д е л е н и е 4. Множество $L \subseteq C$ называется ядром (N) пространства толерантности $\langle C, \tau \rangle$, если существует такая совокупность классов $K_1, K_2, \ldots$, что L — есть совокупность всех элементов из множества C, каждый из которых входит во все эти и только эти классы.

Понятия, введенные в последних трех определениях, иллюстрирует рис. 1.

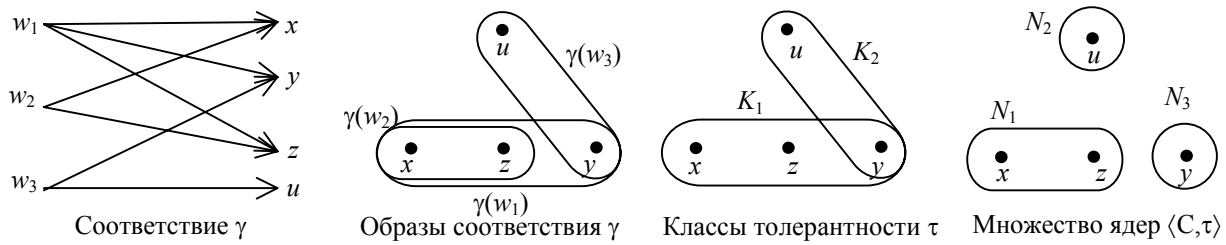


Рис. 1. Пример построения множества ядер пространства толерантности $\langle C, \tau \rangle$

Известно [3], что непустые ядра пространства толерантности $\langle C, \tau \rangle$ образуют разбиение множества C и тем самым единственным образом задают на нем отношение эквивалентности ω в соответствии с правилом: $(x, y) \in \omega \leftrightarrow \gamma^{-1}(x) = \gamma^{-1}(y)$, где x и y — элементы множества C. Таким образом, ядра пространства толерантности $\langle C, \tau \rangle$ представляют собой классы отношения эквивалентности ω на множестве C и при этом элементы каждого такого класса имеют одинаковые прообразы соответствия γ в множестве целей W. В практическом отношении это означает следующее: для того чтобы установить факт соответствия текущего состояния системы некоторой заданной ЦУ, достаточно идентифицировать только класс отношения эквивалентности ω (ядро пространства толерантности $\langle C, \tau \rangle$), которому принадлежит это состояние. При этом мощность множества таких ядер соответствует минимальному числу признаков, позволяющих решить указанную выше задачу.

Переход от использования для этой цели информации о состояниях системы к использованию информации о классах отношения ω позволяет представить задачу идентификации ЦУ в безызбыточном виде. Это обусловлено тем, что множество классов отношения эквивалентности ω может рассматриваться как минимальное множество классификационных признаков элементов множества W для заданного соответствия γ.

Это обстоятельство позволяет конкретизировать правило построения соответствия ϕ_R (1):

$$\forall d \in D_R, \exists x \in W \; d \in \phi_R(x) \leftrightarrow \varphi_R^{-1}(d) \subseteq N_x \wedge x \in \gamma^{-1}(N_x),$$

где N_x — ядро пространства толерантности $\langle C, \tau \rangle$.

По определению множество $\varphi_R^{-1}(d)$ представляет собой класс отношения эквивалентности R, а ядро N_x — класс отношения эквивалентности ω. Оба эти отношения заданы на одном множестве — C. Следовательно, включение $\varphi_R^{-1}(d) \subseteq N_x$ для всех элементов $d \in D_R$ равносильно включению $R \subseteq \omega$, что позволяет окончательно представить правило (1) в виде

$$\forall d \in D_R, \exists x \in W \; d \in \phi_R(x) \leftrightarrow R \subseteq \omega. \qquad (2)$$

Перейдем теперь непосредственно к решению задачи выбора модели, предназначенной для управления ТС при заданном множестве ЦУ. Для этого проанализируем структурные закономерности множества отношений эквивалентности $\Re$, для чего введем на этом множестве отношение строгого порядка „$\succ$“ в соответствии с правилом $R_1 > R_2 \leftrightarrow R_2 \subset R_1$, где R_1 и R_2 — элементы множества $\Re$. Отношение „$\succ$“ является *несовершенным* строгим порядком, так как в общем случае не все элементы множества $\Re$ могут быть сравнимы между собой по этому отношению. В множестве $\Re$ по отношению „$\succ$“ существуют максимальный и минимальный элементы. Максимальному элементу соответствует отношение эквивалентности с единственным классом, совпадающим с множеством C. Классы минимального отношения эквивалентности взаимно-однозначно соответствуют элементам множества C.

Ранее было установлено, что для решения задачи однозначной идентификации целевых состояний ТС должно выполняться условие (2). Очевидно, что из двух любых удовлетворяющих этому условию моделей предпочтительнее выбрать модель с меньшей мощностью множества состояний, так как она будет содержать меньше избыточной информации, необходимой для решения задач управления ТС. Можно показать, что для любого отношения эквивалентности ω, заданного на множестве C, в множестве $\Re$ всегда найдется такой *единственный* элемент R, для которого выполняется следующее правило:

$$\forall R_i \in \Re[R \subseteq \omega \wedge (R > R_i \vee R_i \not\supseteq \omega)] = \text{И}. \qquad (3)$$

Очевидно, что модель, соответствующая отношению R, удовлетворяющему правилу (3), является искомой моделью, содержащей необходимую и достаточную информацию для решения задач управления ТС. Она позволяет осуществлять однозначную идентификацию целей управления системой по состояниям ТС, используемым в этой модели, осуществлять детерминированное управление и контроль поведения системы в пространстве таких состояний [1] и при этом содержит их минимально возможное число, достаточное для осуществления этих функций.

На рис. 2 приведен пример преобразования модели 1 (модели переходов в пространстве состояний 1—6 некоторой системы, которые осуществляются при выполнении некоторых условий $v_1 - v_4$) в модель 2. Обе модели содержат информацию о детерминированных переходах в пространстве состояний системы. При этом если модель 1 содержит информацию о шести состояниях, то модель 2 содержит только три состояния, соответствующие классам отношения эквивалентности R_2, — {5}, {1,3} и {2,4,6}.

На рис. 3 для приведенного примера исходного описания системы (модель 1) представлено множество $\Re$ отношений эквивалентности, элементы которого заданы классами, образованными путем соответствующих объединений состояний модели 1. Анализ показывает,

что для этого примера исходного описания системы может быть построено 13 детерминированных моделей, на множестве которых может быть задано отношение строгого порядка „$\succ$“, показанное стрелками.

Предположим, что в этом примере отношение ω задано на множестве C множеством своих классов — {1}, {3,5}, {2,4,6}. Все тоновые элементы на рис. 3 соответствуют моделям системы, удовлетворяющим условию (2). Условию (3) удовлетворяет единственная модель, состояния которой взаимно-однозначно соответствуют целям управления — {1}, {3,5}, {2,4,6}.

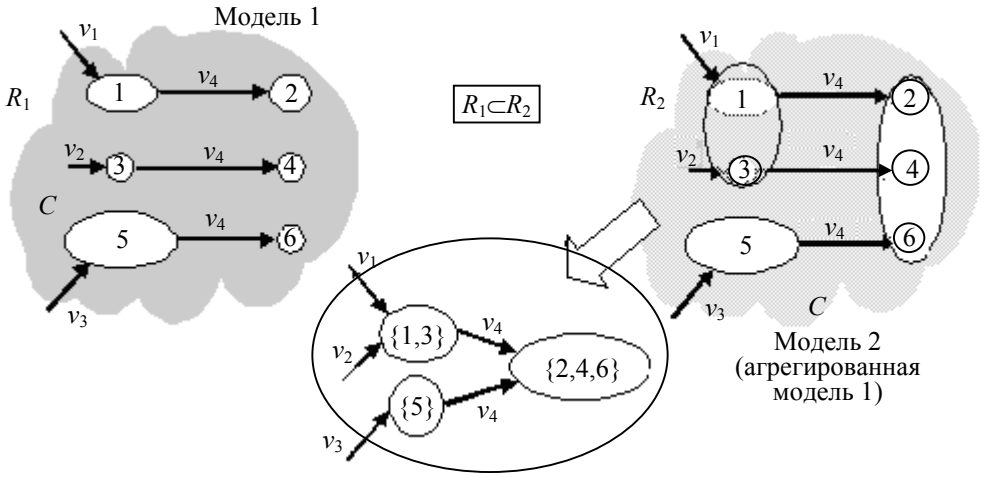


Рис. 2. Пример преобразования модели 1 в модель 2

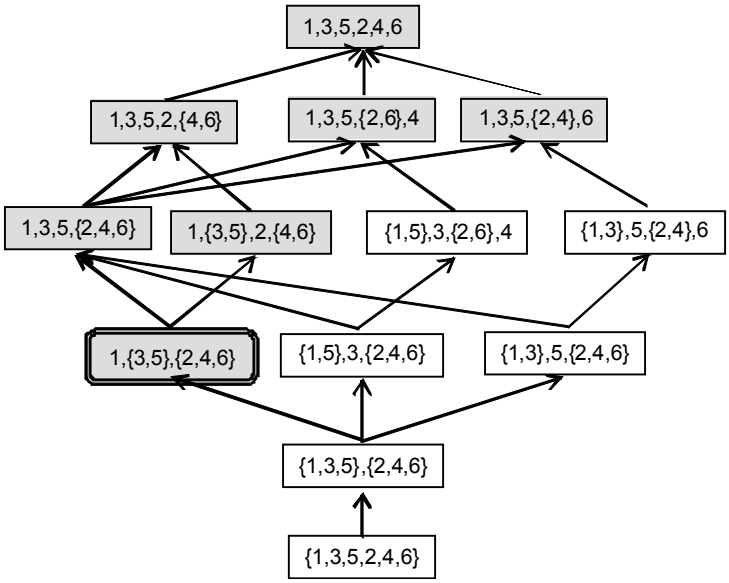


Рис. 3. Отношение строгого порядка на множестве моделей

Рассмотренный теоретический подход к выбору модели, соответствующей целям управления ТС, позволяет сделать важный для построения инженерной методики моделирования ТС вывод: работы по моделированию следует начинать с определения целей управления системой и их семантической интерпретации в терминах значений наблюдаемых параметров (для заданных фиксированных условий функционирования ТС) и только после этого переходить к построению моделей поведения системы в пространстве ее состояний. Такая последовательность работ позволит целенаправленно отбирать и использовать информацию для построения безызбыточных моделей, т.е. избежать непроизводительных усилий на построение моделей, содержащих излишнюю информацию, или моделей, не соответствующих целям управления системой.

Исследования по рассматриваемой тематике проводились при финансовой поддержке Российского фонда фундаментальных исследований (грант 08-08-00346а) и ОИТВС РАН (проект № 2.5), а также Санкт-Петербургского научного центра РАН.

СПИСОК ЛИТЕРАТУРЫ

1. *Кириллов Н. П.* Построение моделей процессов функционирования технических систем по их исходным описаниям // Изв. вузов. Приборостроение. 2006. Т. 49, № 11. С. 12—16.

2. *Эшби У. Р.* Введение в кибернетику. М.: Изд-во иностр. лит., 1959. 455 с.

3. *Шрейдер Ю. А.* Равенство. Сходство. Порядок. М.: Наука, 1971. 256 с.

Сведения об авторе

Николай Петрович Кириллов — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: knp@mail.ru

Поступила в редакцию
06.05.08 г.

УДК 519.8

А. В. ИКОННИКОВА, И. А. ПЕТРОВА, С. А., ПОТРЯСАЕВ, Б. В. СОКОЛОВ

ДИНАМИЧЕСКАЯ МОДЕЛЬ КОМПЛЕКСНОГО ПЛАНИРОВАНИЯ МОДЕРНИЗАЦИИ И ФУНКЦИОНИРОВАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Предложена оригинальная динамическая модель комплексного планирования модернизации и функционирования катастрофоустойчивой информационной системы, позволяющей формально описать основные аспекты и особенности рассматриваемых взаимодействующих процессов.

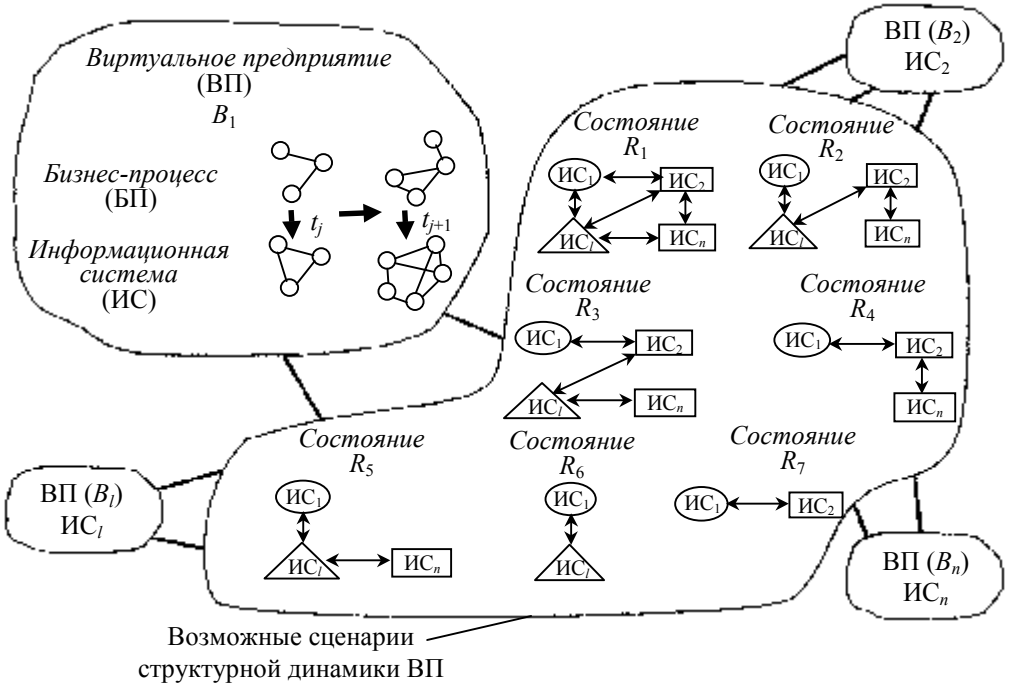
Ключевые слова: *катастрофоустойчивая информационная система, комплексное планирование, модернизация и функционирование сложных объектов.*

Введение. Одной из наиболее перспективных организационных форм современных производственно-транспортных сетей являются виртуальные предприятия (ВП), представляющие собой организации, формируемые из географически распределенных независимых многопрофильных партнеров (реальных предприятий), объединенных на время выполнения совместного заказа в единую организационно-техническую структуру на основе информационно-телекоммуникационных технологий [1—4].

Основное предназначение ВП состоит во временном совместном динамическом использовании различными физическими и юридическими лицами части своих ресурсов в целях получения каждым из них прибыли в ходе решения общей производственной задачи. Важнейшей подсистемой любого ВП является его интегрированная информационная система (ИС), образованная на основе оперативного конфигурирования (структурно-функционального синтеза) ИС, обеспечивающих функционирование как отдельного реального предприятия, входящего во временную кооперацию, так и их взаимодействие в процессе производственной деятельности. Необходимо отметить, что для интегрированной ИС (как и для ВП) характерна структурная динамика, вызванная различными причинами [4, 5].

На рисунке в графическом виде представлены возможные варианты сценариев структурной динамики применительно к современным ИС. Особую опасность для современных ИС представляют причины, которые приводят к возникновению кризисных ситуаций, аварий

и катастроф, имеющих природно-экологический, технико-производственный или антропогенно-социальный характер. При этом спектр угроз экономической, физической и информационной безопасности, а также перечень уязвимостей аппаратно-программной и информационной инфраструктур ИС постоянно расширяется [4].



Состав и структура интегрированной информационной системы виртуального предприятия

Кроме того, в реальной жизни возможны ситуации, когда указанные угрозы являются комбинированными и приводят к лавинообразному возникновению и развитию негативных событий, вызывающих в конечном итоге катастрофические последствия. В этих условиях обеспечение непрерывности бизнес-процессов (БП) и повышение катастрофоустойчивости соответствующих производственных систем (бизнес-систем — БС), входящих в состав ВП, является одним из важнейших стратегических направлений развития современной экономики. При этом под катастрофоустойчивостью ИС следует понимать способность компьютерного комплекса, состоящего из нескольких систем, сохранять критически важные данные и обеспечивать выполнение своих основных функций после массового (возможно, целенаправленного) уничтожения его компонентов в результате различных катаклизмов как природного характера, так и инспирированных человеком [4, 5].

Перечисленные особенности катастрофоустойчивых информационных систем (КУИС) приводят к необходимости с принципиально новых позиций подходить к решению проблем управления развитием указанных систем, для которых переход от „старого“ (существующего) варианта ИС к „новому“ (модернизированному либо восстанавливаемому) варианту КУИС не может быть проведен мгновенно. На практике это приводит к тому, что на достаточно длительном интервале времени (периоде модернизации КУИС либо восстановления ее работоспособности) осуществляется совместная эксплуатация элементов и подсистем „старой“ и „новой“ КУИС. Однако в этих условиях показатели качества и эффективности бизнес-процессов, поддерживаемых данными КУИС, не должны ухудшаться. Таким образом, всякое изменение и развитие той или иной подсистемы (структуры) КУИС объективно осуществляется одновременно с решением оперативных (текущих) задач, стоящих перед соответствующей БС. Поэтому возникает необходимость совместной постановки задач комплексного планирования модернизации и функционирования КУИС. Базируясь на результатах предыдущих

исследований, приведем упрощенный вариант комплексной динамической модели программного управления модернизацией и функционированием КУИС [4, 5]. При этом процессы модернизации КУИС будем трактовать широко, интерпретируя их также (в зависимости от складывающейся обстановки) и как процессы восстановления работоспособности КУИС после аварии или нештатной ситуации.

Представленные в настоящей статье математические модели базируются на результатах, полученных в ранее опубликованных работах по рассматриваемой тематике [5, 6].

Для того чтобы формально описать рассматриваемые модели введем, прежде всего, следующие множества: $A^{(\text{о},j)} = \left\{ A_\nu^{(\text{о},j)};\ \nu = 1,...,n_j \right\}$ — множество БП, выполняемых в узле B_j ВП; $A^{(\text{л},j)} = \left\{ A_\nu^{(\text{л},j)};\ \nu = 1,...,n_j \right\}$ — множество технологий обработки информационных потоков, реализуемых в узле B_j, для обеспечения выполнения соответствующих БП; $B = \left\{ B_j;\ j = 1,...,m \right\}$ — множество подсистем (узлов) ВП; $D_\nu^{(\text{л},j)} = \left\{ D_{\nu\chi}^{(\text{л},j)};\ \chi = 1,...,S_\nu \right\}$ — множество операций, входящих в технологию $A_\nu^{(\text{л},j)}$, предназначенную для обработки информации, необходимой для реализации бизнес-процесса $A_\nu^{(\text{о},j)}$; $B^{(\text{л},j)} = \left\{ B_r^{(\text{л},j)};\ r = 1,...,R_j \right\}$ — множество информационных ресурсов, модернизируемых в узле B_j ВП; $B^{(\text{р},j)} = \left\{ B_\delta^{(\text{р},j)};\ \delta = 1,...,\Delta_j \right\}$ — множество материальных ресурсов, выделяемых для модернизации информационных ресурсов; $D_r^{(\text{р},j)} = \left\{ D_{rk}^{(\text{р},j)};\ k = 1,...,\text{л}_j^{(r)} \right\}$ — множество операций, входящих в технологический цикл управления (ТЦУ) модернизацией информационного ресурса $B_r^{(\text{л},j)}$ в узле B_j ВП.

Описание динамической модели. Рассмотрим *математическую модель программного управления функционированием КУИС в узле B_j ВП.*

Математическая модель процесса представляется выражениями

$$\frac{dx_{\nu\chi}^{(\text{л},j)}}{dt} = \sum_{r=1}^{R_j} u_{\nu\chi r}^{(\text{л},j)};\ \ \frac{dx_r^{(\text{л},j)}}{dt} = \sum_{\nu=1}^{n_j} \sum_{\chi=1}^{S_\nu} w_{\nu\chi r}^{(\text{л},j)},\ \ \frac{d\tilde{x}_{rS_\nu}^{(\text{л},j)}}{dt} = \tilde{\omega}_{rS_\nu}^{(\text{л},j)}. \tag{1}$$

Ограничения на управляющие воздействия определяются следующим образом:

$$0 \le u_{\nu\chi r}^{(\text{л},j)}(t) \le \left[e_{\nu\chi r} \left(1 - v_r^{(\text{р},2)}\right) + \overline{e}_{\nu\chi r} v_r^{(\text{р},2)} \right] w_{\nu\chi r}^{(\text{л},j)}; \tag{2}$$

$$\sum_{\nu=1}^{n_j} \sum_{\chi=1}^{S_\nu} V_{\nu\chi}^{(j)} w_{\nu\chi r}^{(\text{л},j)} \le V_r^{(j)} \left(1 - v_r^{(\text{р},2)}\right) + \overline{V}_r^{(j)} v_r^{(\text{р},2)}; \tag{3}$$

$$\sum_{\nu=1}^{n_j} \sum_{\chi=1}^{S_\nu} u_{\nu\chi r}^{(\text{л},j)}(t) \le \Phi_r^{(j)} \left(1 - v_r^{(\text{р},2)}\right) + \overline{\Phi}_r^{(j)} v_r^{(\text{р},2)};\ \ \tilde{\omega}_{rS_\nu}^{(\text{л},j)} \left(a_{\nu S_\nu}^{(\text{л},j)} - x_{\nu S_\nu}^{(\text{л},j)} \right) = 0; \tag{4}$$

$$\sum_{r=1}^{R_j} w_{\nu\chi r}^{(\text{л},j)} \left(a_{\nu(\chi-1)}^{(\text{л},j)} - x_{\nu(\chi-1)}^{(\text{л},j)} \right) = 0, \tag{5}$$

$$\sum_{r=1}^{R_j} w_{\nu\chi r}^{(\text{л},j)}(t) \le 1,\ \forall\chi,\ \forall\nu;\ \ 0 \le w_{\nu\chi r}^{(\text{л},j)}(t) \le 1. \tag{6}$$

Краевые условия описываются выражениями

$$x_{\nu\chi}^{(\text{л},j)} \left(t_0^{(j)} \right) = 0;\ \ x_r^{(\text{л},j)} \left(t_0^{(j)} \right) = 0;\ \ x_{\nu\chi}^{(\text{л},j)} \left(t_f^{(j)} \right) = a_{\nu\chi}^{(\text{л},j)};\ \ x_r^{(\text{л},j)} \left(t_f^{(j)} \right) \in R^1, \tag{7}$$

а показатели качества процесса управления — соотношениями

$$J_1^{(\text{л},j)} = \sum_{r=1}^{R_{j-1}} \sum_{r_1=r+1}^{R_j} \int_{t_0^{(j)}}^{t_f^{(j)}} \left(x_r^{(\text{л},j)}(\tau) - x_{r_1}^{(\text{л},j)}(\tau) \right)^2 d\tau; \tag{8}$$

$$J_2^{(\text{л},j)} = \sum_{\nu=1}^{n_j} \sum_{\chi=1}^{S_\nu} \sum_{r=1}^{R_j} \int_{t_{f0}^{(j)}}^{t_f^{(j)}} \alpha_{\nu\chi r}^{(\text{л},j)}(\tau) w_{\nu\chi r}^{(\text{л},j)}(\tau) d\tau; \tag{9}$$

$$J_3^{(\text{л},j)} = \frac{1}{2} \sum_{\nu=1}^{n_j} \sum_{\chi=1}^{S_\nu} \left(a_{\nu\chi}^{(\text{л},j)} - x_{\nu\chi}^{(\text{л},j)} \left(t_f^{(j)} \right) \right)^2; \; J_4^{(\text{л},j)} = \sum_{r=1}^{R_j} \left(T^{(j)} - x_r^{(\text{л},j)} \left(t_f^{(j)} \right) \right)^2. \tag{10}$$

В соотношениях (1)—(10) переменные интерпретируются следующим образом: $x_{\nu\chi}^{(\text{л},j)}$ — переменная, характеризующая состояние выполнения операции $D_{\nu\chi}^{(\text{л},j)}$ (т.е. текущий объем обработанной информации при выполнении операции $D_{\nu\chi}^{(\text{л},j)}$); $a_{\nu\chi}^{(\text{л},j)}$ — заданный объем информации, который обрабатывается при выполнении операции $D_{\nu\chi}^{(\text{л},j)}$, входящей в состав технологии $A_\nu^{(\text{л},j)}$ обработки информационных потоков для обеспечения выполнения БП $A_\nu^{(\text{о},j)}$; $x_r^{(\text{л},j)}$ — переменная, текущее значение которой численно равно общей продолжительности задействования ресурса $B_r^{(\text{л},j)}$ информационной системы, входящей в состав узла B_j ВП; $u_{\nu\chi r}^{(\text{л},j)}$ — интенсивность обработки на ресурсе $B_r^{(\text{л},j)}$ информации, необходимой для выполнения операции $D_{\nu\chi}^{(\text{о},j)}$, входящей в состав БП $A_\nu^{(\text{о},j)}$; $w_{\nu\chi r}^{(\text{л},j)}$ — управляющее воздействие, принимающее значение 1, если ресурс $B_r^{(\text{л},j)}$ ИС в узле B_j выделяется для выполнения операции $D_{\nu\chi}^{(\text{л},j)}$, в противоположном случае $w_{\nu\chi r}^{(\text{л},j)}(t) = 0$; $e_r^{(j)}$, $V_r^{(j)}$, $\Phi_r^{(j)}$ — заданные величины, характеризующие соответственно максимально возможную интенсивность выполнения операции $D_{\nu\chi}^{(\text{л},j)}$ на ресурсе $B_r^{(\text{л},j)}$, максимально возможный объем доступной оперативной памяти ИС в узле B_j и максимально возможную производительность ресурса $B_r^{(\text{л},j)}$ до его модернизации; $\overline{e}_r^{(j)}$, $\overline{V}_r^{(j)}$, $\overline{\Phi}_r^{(j)}$ — величины, имеющие аналогичную интерпретацию, но соответствующие ситуации, когда модернизация (либо восстановление работоспособности) проведена; $v_{\chi r}^{(\text{р},2)}(t)$ — вспомогательное управляющее воздействие, принимающее значение 1 в момент времени t, если осуществлен переход от „старых“ $\left(e_r^{(j)}, V_r^{(j)}, \Phi_r^{(j)} \right)$ к „новым“ $\left(\overline{e}_r^{(j)}, \overline{V}_r^{(j)}, \overline{\Phi}_r^{(j)} \right)$ информационным ресурсам в узле B_j; $V_{\nu\chi}^{(j)}$ — объем оперативной памяти, которая выделяется для выполнения операции $D_{\nu\chi}^{(\text{л},j)}$ обработки информации; $\alpha_{\nu\chi r}^{(\text{л},j)}(\tau)$ — заданная функция, определяющая качество выполнения соответствующих операций; $\tilde{\omega}_{rS_\nu}^{(\text{л},j)}$ — вспомогательное управляющее воздействие, принимающее значение 1, если реализована полностью технология обработки информации для осуществления БП $A_\nu^{(\text{о},j)}$, 0 — в противном случае.

Ограничения (2)—(4) определяют возможности по переработке информации на ресурсе $B_r^{(\text{л},j)}$. Ограничения (5) определяют очередность выполнения операций $D_{\nu\chi}^{(\text{л},j)}$, $D_{\nu(\chi-1)}^{(\text{л},j)}$, связанных с обработкой информации и необходимых для выполнения соответствующих

операций $D_{\nu\chi}^{(\text{о},j)}$, $D_{\nu(\chi-1)}^{(\text{о},j)}$, входящих в состав БП $A_{\nu}^{(\text{о},j)}$ и $A_{\nu-1}^{(\text{о},j)}$. Ограничения (6) означают, что в текущий момент времени операция $D_{\nu\chi}^{(\text{л},j)}$ может выполняться только на одном информационном ресурсе $B_r^{(\text{л},j)}$, $r = 1,...,R_j$. Соотношения (7) задают краевые условия (ограничения на значения переменных $x_{\nu\chi}^{(\text{л},j)}$, $x_r^{(\text{л},j)}$ в начальный и конечный моменты времени $t_0^{(j)}$ и $t_f^{(j)}$). Показатель (8) предназначен для оценивания степени равномерности использования ресурсов $B_r^{(\text{л},j)}$, $B_{r_1}^{(\text{л},j)}$, $r, r_1 = 1,...,R_j$. Показатель (9) позволяет оценить суммарное качество выполнения всей совокупности операций $D_{\nu\chi}^{(\text{л},j)}$ при фиксированной программе их выполнения $w_{\nu\chi r}^{(\text{л},j)}(\tau)$. Показатели (10) вводятся в том случае, если необходимо оценить точность выполнения краевых условий (7) либо минимизировать потери, вызванные невыполнением операции $D_{\nu\chi}^{(\text{л},j)}$.

Рассмотрим *математическую модель программного управления модернизацией элементов КУИС в узле B_j ВП.*

Математическая модель процесса представляется выражениями

$$\frac{dx_{rk}^{(\text{р},1)}}{dt} = \sum_{\delta=1}^{\Delta j} b_{r\delta k} v_{r\delta k}^{(\text{р},1)}, \quad \frac{dx_r^{(\text{р},2)}}{dt} = v_r^{(\text{р},2)}. \tag{11}$$

Ограничения на управляющие воздействия определяются как

$$\sum_{r=1}^{R_j} v_{r\delta k}^{(\text{р},1)}(t) \le c_{\delta j}^{(\text{р},1)}, \ \forall\delta, \ \forall k; \tag{12}$$

$$\sum_{\delta=1}^{\Delta j} v_{r\delta k}^{(\text{р},1)}(t) \le 1, \ \forall\delta, \ \forall k, k = 1,...,\text{л}_j^{(r)}; \tag{13}$$

$$\sum_{\delta=1}^{\Delta j} v_{r\delta k}^{(\text{р},1)}\left(a_{r(k-1)}^{(\text{р},1)} - x_{r(k-1)}^{(\text{р},1)}\right) = 0; \tag{14}$$

$$v_r^{(\text{р},2)}\left(a_{r\text{л}_j}^{(\text{р},1)} - x_{r\text{л}_j}^{(\text{р},1)}\right) = 0; \tag{15}$$

$$0 \le v_{r\delta k}^{(\text{р},1)}(t) \le 1; \ 0 \le v_r^{(\text{р},2)}(t) \le 1. \tag{16}$$

Краевые условия задаются выражениями

$$x_{rk}^{(\text{р},1)}\left(t_0^{(j)}\right) = 0; \ x_r^{(\text{р},2)}\left(t_0^{(j)}\right) = 0; \ x_{rk}^{(\text{р},1)}\left(t_f^{(j)}\right) = a_{rk}^{(\text{р},1)}; \ x_r^{(\text{р},2)}\left(t_f^{(j)}\right) \in R^1, \tag{17}$$

а показатели качества процесса управления — соотношением

$$J_1^{(\text{р},1)} = \sum_{\delta=1}^{\Delta j}\sum_{r=1}^{R_\nu}\sum_{k=1}^{\text{л}_j}\left[\left(\int_{t_0^{(j)}}^{t_f^{(j)}}\left(\lambda_1^{(3)} c_{r\delta k}^{(\text{р},j)}(\tau) + \lambda_2^{(3)}\beta_{rk}^{(\text{р},j)}(\tau)\right)v_{r\delta k}^{(\text{р},1)} d\tau\right) + \lambda_3^{(3)}\frac{1}{2}\left(a_{rk}^{(\text{р},1)} - x_{rk}^{(\text{р},1)}\left(t_f^j\right)\right)^2\right]. \tag{18}$$

В соотношениях (11)—(18) переменные интерпретируются следующим образом: $x_{rk}^{(\text{р},1)}(t)$ — переменная, характеризующая текущее состояние выполнения операции $D_{rk}^{(\text{р},j)}$, связанной с модернизацией информационного ресурса $B_r^{(\text{л},j)}$ в узле B_j; $a_{rk}^{(\text{р},1)}(t)$ — заданный

объем выполнения операции $D_{rk}^{(\text{р},j)}$, входящей в состав технологического цикла управления $D_r^{(\text{р},j)}$ модернизацией информационного ресурса $B_r^{(\text{л},j)}$ в узле B_j; $a_{r\text{л}_j}^{(\text{р},1)}$ — заданный объем выполнения последней операции $D_{r\text{л}_j}^{(\text{р},j)}$, входящей в ТЦУ модернизацией информационного ресурса $B_r^{(\text{л},j)}$ в узле B_j; $v_{r\delta k}^{(\text{р},1)}(t)$ — управляющее воздействие, принимающее значение 1, если на выполнение операции $D_{rk}^{(\text{р},j)}$, связанной с модернизацией информационного ресурса $B_r^{(\text{л},j)}$, в данный момент выделен ресурс $B_\delta^{(\text{р},j)}$; в противоположном случае $v_{r\delta k}^{(\text{р},2)}(t)=0$; $v_r^{(\text{р},2)}(t)$ — вспомогательное управляющее воздействие, принимающее значение 1, если процесс модернизации информационного ресурса $B_r^{(\text{л},j)}$ завершился; в противоположном случае $v_r^{(\text{р},2)}(t)=0$; $x_r^{(\text{р},2)}(t)$ — переменная, характеризующая текущее состояние выполнения вспомогательной операции: ее значение численно равняется величине временного интервала, прошедшего с момента окончания модернизации информационного ресурса $B_r^{(\text{л},j)}$ до момента окончания интервала планирования; $b_{r\delta}^{(k)}$ — интенсивность выполнения операции $D_{rk}^{(\text{р},j)}$, связанной с модернизацией ресурса $B_r^{(\text{л},j)}$.

Ограничения (12) определяют возможность одновременного выполнения $c_{\delta j}^{(\text{р},1)}$ операций вида $D_{rk}^{(\text{р},j)}$, входящих в состав ТЦУ $D_r^{(\text{р},j)}$ модернизацией информационного ресурса $B_r^{(\text{л},j)}$. Ограничения (13) задают требование о том, что операция $D_{rk}^{(\text{р},j)}$, входящая в состав ТЦУ $D_r^{(\text{р},j)}$ модернизацией информационного ресурса $B_r^{(\text{л},j)}$, сама может выполняться только с использованием одного из ресурсов $B_\delta^{(\text{р},j)}$, выделенных на модернизацию. Ограничения (14) определяют очередность выполнения операций $D_{rk}^{(\text{р},j)}$ и $D_{r(k-1)}^{(\text{р},j)}$, связанных с модернизацией информационного ресурса $B_r^{(\text{л},j)}$. Ограничения (15) определяют условия окончания процесса модернизации информационного ресурса $B_r^{(\text{л},j)}$ в узле B_j ВП. Ограничения (16) задают область изменения возможных значений управляющих воздействий $v_{r\delta k}^{(\text{р},1)}(t)$ и $v_r^{(\text{р},2)}(t)$. Соотношения (17) задают ограничения на значения переменных $x_{rk}^{(\text{р},1)}(t)$ и $x_r^{(\text{р},2)}(t)$ в моменты времени $t_0^{(j)}$ и $t_f^{(j)}$ (моменты начала и окончания интервала планирования модернизации информационной системы в узле B_j ВП).

Показатель качества процесса управления модернизацией вида (18) позволяет одновременно оценить как суммарные стоимостные затраты на проведение модернизации и текущую эксплуатацию КУИС в узле B_j, так и суммарный штраф за нарушение директивных сроков выполнения операций, входящих в ТЦУ $D_r^{(\text{р},j)}$ модернизацией информационных ресурсов $B_r^{(\text{л},j)}$. В соотношении (18) весовые коэффициенты $\lambda_1^{(3)}$, $\lambda_2^{(3)}$, $\lambda_3^{(3)}$ считаются известными величинами. Указанные величины можно подсчитать с использованием первого компонента показателя (18). Терминальный (второй) компонент показателя (18) вводится для оценивания точности выполнения краевых условий (17) либо минимизации потерь, связанных с невыполнением операций $D_{rk}^{(\text{р},j)}$, входящих в ТЦУ $D_r^{(\text{р},j)}$ модернизацией информационного ресурса $B_r^{(\text{л},j)}$.

Заключение. Анализ программной реализации предложенного варианта формального описания процессов модернизации и функционирования КУИС показывает, что представленные динамические модели обладают следующими основными достоинствами:

— позволяют широко использовать в ходе планирования фундаментальные научные результаты, полученные к настоящему времени в современной теории управления сложными динамическими системами с перестраиваемой структурой;

— позволяют существенно сократить размерность задач планирования, решаемых в каждый момент времени;

— предоставляют возможность конструктивно проводить согласование и взаимную интерпретацию результатов, полученных на аналитических и имитационных моделях планирования как на концептуальном, так и на алгоритмическом, информационном, программном уровнях описания;

— позволяют обоснованно подходить к выбору временных интервалов работы элементов и подсистем КУИС;

— существенно сокращают затраты оперативной памяти ЭВМ, повышают оперативность решения задач планирования при использовании перспективных гибридных вычислительных систем, позволяющих проводить декомпозицию и распараллеливание вычислительного процесса.

Представленная детерминированная динамическая модель планирования может быть дополнена ранее разработанной комбинированной моделью управления структурной динамикой информационной системы [6]. В этом случае появляется возможность не только планировать процессы модернизации (восстановления) и функционирования интегрированной ИС ВП в каждом отдельном узле, но и осуществлять программное управление соответствующей телекоммуникационной системой с учетом ее структурной динамики.

К настоящему времени разработано несколько версий прототипа программного обеспечения решения задач рассматриваемого класса применительно к различным предметным областям (космонавтика, энергетика, менеджмент), которые подтвердили работоспособность и эффективность предложенного модельно-алгоритмического обеспечения.

Исследования, выполненные по данной тематике, проводились при финансовой поддержке Российского фонда фундаментальных исследований (гранты № 07–07–00169, 06–07–89242, 08–08–00403), Отделения нанотехнологий и информационных технологий РАН (проект № О–2.5/03).

СПИСОК ЛИТЕРАТУРЫ

1. Virtual Enterprises and Collaborative Networks // Ed. *L. Camarihna-Matos*. Berlin: Kluwer Academic Publishers, 2004.

2. *Wang L., Norrie D. H.* Process planning and control in a holonic manufacturing enviroment // J. of Applied Systems Studies. 2001. N 2(1). P. 106—126.

3. *Иванов Д. А.* Виртуальные предприятия и логистические цепи: комплексный подход к организации и оперативному управлению в новых формах производственной кооперации. СПб.: Изд-во СПбГУЭФ, 2003.

4. *Будзко В. И., Беленков В. Г., Кейер П. А.* К выбору варианта построения катастрофоустойчивых информационно-телекоммуникационных систем // Системы и средства информатики. 2003. Вып. 13. С. 16—40.

5. *Охтилев М. Ю., Соколов Б. В., Юсупов Р. М.* Интеллектуальные технологии мониторинга и управления структурной динамикой сложных технических систем. М.: Наука, 2006.

6. *Москвин Б. В., Михайлов Е. П., Павлов А. Н., Соколов Б. В.* Комбинированные модели управления структурной динамикой информационных систем // Изв. вузов. Приборостроение. 2006. Т. 49, № 11. С. 7—11.

Сведения об авторах

Анна Владимировна Иконникова — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: ikonnikova@iias.spb.su

Ирина Андреевна Петрова — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: irina.petrova.9@gmail.com

Семен Алексеевич Потрясаев — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: semp@mail.ru

Борис Владимирович Соколов — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: sokol@iias.spb.su

Поступила в редакцию
06.05.08 г.

УДК 681.3.06

В. В. Михайлов, И. С. Селяков

ИСПОЛЬЗОВАНИЕ МУЛЬТИАГЕНТНОГО СИМУЛЯТОРА ПРИ МОДЕЛИРОВАНИИ РАСПРЕДЕЛЕННЫХ СИСТЕМ

Рассматриваются особенности применения многоагентного подхода при моделировании сложных распределенных систем. Приводятся структуры мультиагентного симулятора и многоагентной модели, лежащей в его основе, а также принципы стыковки симулятора с геоинформационной системой с использованием Shape-файлов. Предложенный программный пакет позволяет решать задачи в разных предметных областях, в том числе исследовать пространственно-временную динамику сообществ и популяций живых организмов с точной географической привязкой их территориального размещения.

Ключевые слова: *мультиагентные системы, распределенная модель, симулятор, графическое представление данных, геоинформационные системы.*

Введение. Мультиагентные системы [1] представляют новую область знаний и являются универсальными для решения задач в таких областях, как экономика, экология, логистика, мониторинг бизнес-процессов и др. Учет пространственной специфики и переход к распределенным моделям при использовании многоагентного подхода осуществляется естественным путем, когда каждый агент, характеризуемый определенным набором свойств, занимает свое положение в пространстве, входит в состав сообществ, миграционных потоков и т.п. Многоагентный подход рассматривается как наиболее прогрессивный для разработки и анализа сложных интеллектуальных распределенных информационных систем.

Преимущество многоагентного подхода для моделирования экологических систем заключается в возможности одновременного моделирования динамики численности и пространственного размещения сообществ и популяций, состоящих из большого числа относительно независимых и сходных по своим характеристикам объектов.

Важным аспектом при моделировании является наглядное представление перемещения агентов в пространстве, их взаимодействия друг с другом и с окружающей средой. В настоящее время существует отдельная область знаний, занимающаяся созданием графического представления и отображения пространственных данных, — географические информационные системы (ГИС). Для представления пространственного размещения агентов на территории применение методов и современных программных средств ГИС является наиболее продуктивным.

В настоящей статье описывается структура разработанной многоагентой модели, используемой для моделирования распределенных систем, например популяции животных. Также представлен графический мультиагентный симулятор, который реализует многоагентную

модель и осуществляет необходимые вычисления. Кроме того, в статье рассматривается механизм взаимодействия симулятора с ГИС для графического представления пространственно-временной динамики агентов на заданной территории.

Структура многоагентной модели распределенной системы. Объекты (или группы объектов) в системе представлены отдельными агентами, реализующими определенную модель поведения. При моделировании популяции животных различают агентов, представляющих мужские и женские особи. В системе также вводится агент-координатор, в задачи которого входит определение общего времени для синхронизации деятельности агентов, контроль жизненного цикла отдельных агентов и предоставление им определенных сервисов.

Агенты обмениваются информацией с координатором по методу „запрос — ответ“. Каждый агент, определив в системе агента-координатора, посылает запрос на регистрацию. Агент-координатор, получив запросы от нескольких агентов, подтверждает их регистрацию и отправляет всем первый синхроимпульс, в результате чего агенты совершают „шаг“ в системе, изменяя тем самым свои координаты. При этом общее время внутри системы увеличивается на шаг счета. Получив синхроимпульс и выполнив все необходимые действия, каждый агент отправляет координатору подтверждение. Координатор, в свою очередь, рассылает новый синхроимпульс, получив подтверждения ото всех агентов, а также постоянно обновляет базу данных с информацией обо всех агентах в системе.

Мультиагентый симулятор. Для реализации многоагентной модели в лаборатории информационных технологий в системном анализе и моделировании СПИИРАН был разработан симулятор, предназначенный, в том числе, для моделирования популяций животных.

В области моделирования агентных систем имеется множество стандартов и подходов к построению моделей. В основном это связано с тем, что многоагентное моделирование является относительно новой областью знаний, где еще не до конца выработаны единые стандарты. Проводимые в последнее время исследования в этом направлении привели к появлению таких стандартов, как FIPA [2] и OMG.

При поиске оптимального программного пакета для реализации многоагентной системы были рассмотрены пакеты FIPA-OS [3], JADE [4, 5], Zeus [6], TAEMS [7], AnyLogic [8] и MASDK [9]. По результатам исследования в качестве оптимального был выбран пакет JADE (Java Agent Development Framework), созданный при участии нескольких университетов и научных групп. Этот пакет полностью поддерживает стандарт для мультиагентных систем FIPA, который на сегодняшний день является передовым в области многоагентного моделирования.

Важной особенностью пакета JADE является тот факт, что он полностью написан на языке Java. По сути, JADE — это набор Java-библиотек, которые предоставляют программисту определенный интерфейс. Этот пакет позволяет создавать агентов, поддерживать их жизненный цикл, назначать агентам задачи, определять их поведение и т.п. Таким образом, JADE реализует парадигму так называемого агентно-ориентированного программирования. Пакет JADE не имеет встроенной графической поддержки, за исключением специальных отладочных средств. Другим важным моментом является многоплатформенность языка Java. Это означает, что симулятор может одинаково успешно работать с разными операционными системами и на различных по архитектуре устройствах. Во многом благодаря этому разработка симулятора осуществлялась средствами языка Java и библиотек JADE.

На рис. 1 представлено окно монитора, отображающее мультиагентный симулятор. Слева расположена панель управления процессом моделирования, с помощью которой можно запускать и приостанавливать работу модели. Также возможен режим пошагового моделирования, когда решение выполняется с остановками на каждом шаге. Справа расположена рабочая область, где отображается перемещение агентов в пространстве, — в каждой ячейке поля может содержаться несколько агентов различных типов (например, мужские и женские особи).

В верхней части главного окна расположено меню, с помощью которого задаются начальные условия, определяются параметры моделирования и размеры координатной плоскости. Моделирование можно проводить как в детерминированном, так и вероятностном режиме.

Результаты моделирования могут быть обобщены и представлены в табличной или графической форме. В частности, в дополнительном окне может быть построен график (рис. 2), отражающий изменение численности (N) популяции животных в зависимости от времени (кривая *1* соответствует численности самок, кривая *2* — численности самцов).

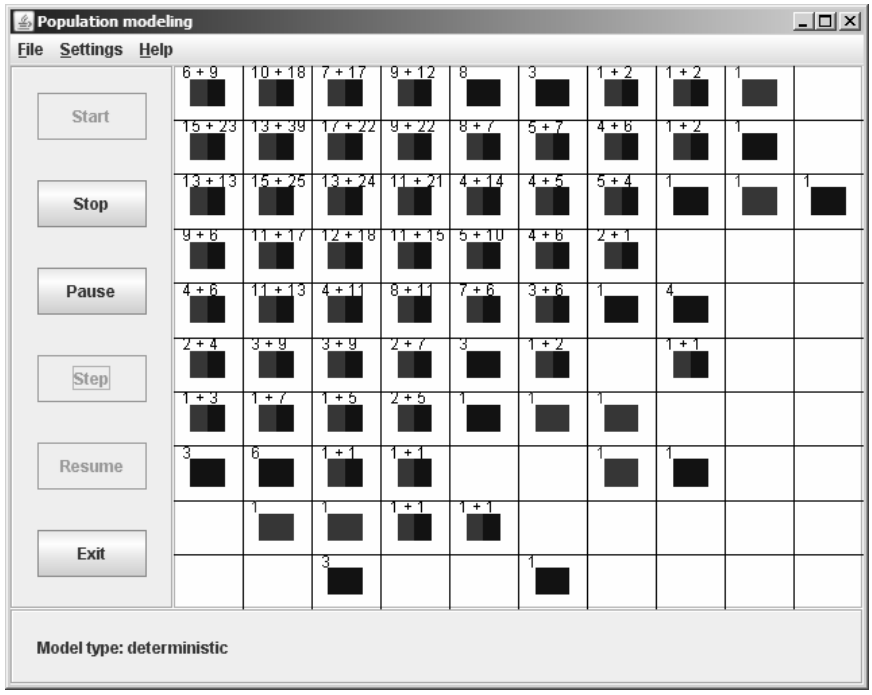


Рис. 1. Экранная форма мультиагентного симулятора

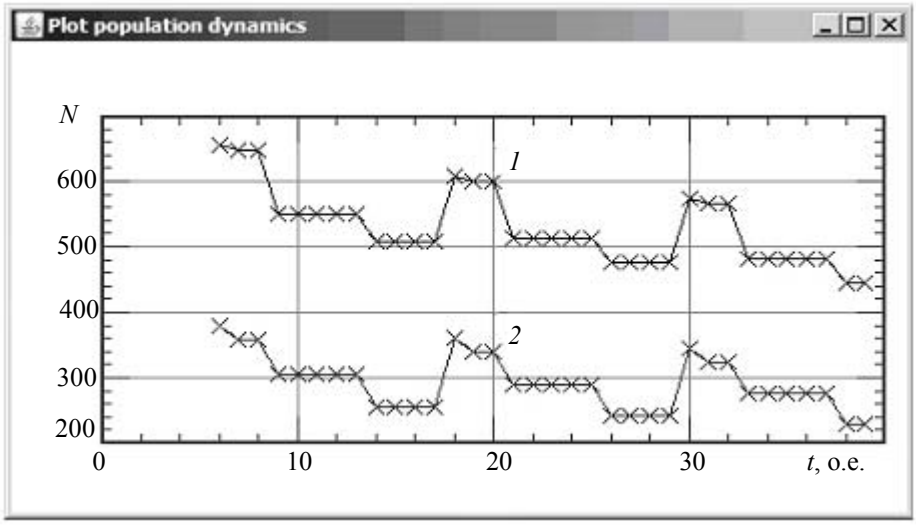


Рис. 2. Графическое окно мультиагентного симулятора

Следует отметить, что в основном окне симулятора перемещение агентов в пространстве показано достаточно упрощенно. Такое представление не дает информации об особенностях среды, в которой находятся агенты. Поэтому была поставлена задача стыковки симулятора с ГИС для более наглядного представления пространственно-временной динамики распределенной системы.

Взаимодействие симулятора с ГИС. Геоинформационные системы предназначены для сбора, хранения, анализа и графической визуализации пространственных данных и связанной с ними информации о представленных в системе объектах.

Среди многообразия ГИС одним из наиболее популярных и универсальных пакетов является ArcGIS [10]. Этот пакет позволяет создавать и редактировать карты, добавлять к ним отдельные тематические слои, хранить данные в различных современных базах данных, отображать пространственные данные на картах.

Пакет ArcGIS предоставляет пользователю возможность хранить данные об отдельных тематических слоях карты разными способами. Один из них — хранение данных в виде так называемого Shape-файла. Формат этого файла является стандартом для хранения пространственной информации в других ГИС. Именно этот способ представления данных был применен при организации взаимодействия мультиагентного симулятора с ГИС.

Схема работы пользователя с симулятором при использовании ГИС представлена ниже:

— на начальном этапе пользователь определяет размещение агентов на территории средствами любой ГИС, которая способна сохранять данные в формате Shape-файла;

— пользователь задает также дополнительные параметры агентов (например, пол и возраст особей при моделировании популяций) в виде атрибутов;

— мультиагентный симулятор переводится в режим работы с ГИС через главное меню, при этом созданный Shape-файл задает начальные условия для моделирования;

— в режиме моделирования симулятор производит вычисления и обновляет данные о размещении агентов в Shape-файле, при этом в основном окне симулятора никакая информация не отображается;

— ГИС автоматически вносит дополнительные данные из Shape-файла в реальном времени и наносит их на карту, позволяя пользователю наблюдать за перемещением агентов на конкретной территории с точной географической привязкой.

Для обработки данных в Shape-файлах использовалась Java-библиотека GeoTools [11], имеющая богатый набор интерфейсов для работы с данными ГИС.

В качестве системы отображения пространственного размещения агентов можно, например, воспользоваться пакетом ArcGIS Explorer [12]. Этот пакет способен отображать пространственные данные, хранящиеся в разных форматах, в том числе в Shape-файлах. ArcGIS Explorer имеет большое количество встроенных подробных карт земного шара, на которые наносятся данные о размещении агентов. В настройках пакета можно задать тип карты, способ отображения, а также время обновления информации.

Заключение. Моделирование сложных распределенных систем может осуществляться с использованием различных схем. Примером распределенной системы является популяция животных. Применение агентного подхода к моделированию такой системы позволяет на основе знаний о поведении отдельных особей имитировать процесс формирования группировок и их перемещений в пределах ареала как целостных образований.

Рассмотренный в настоящей статье многоагентный симулятор реализует многоагентную модель и предоставляет пользователю возможность проводить различные компьютерные эксперименты, например, по исследованию особенностей размещения и миграций животных с учетом территориальной неоднородности условий их обитания. Решена задача стыковки многоагентного симулятора с ГИС, что позволяет получить наглядное представление о размещении агентов на конкретной территории с точной географической привязкой. Использование Shape-файлов обеспечивает при этом работу симулятора с ГИС в реальном масштабе времени.

Разработанная система используется в настоящее время для выявления закономерностей территориального размещения и миграции животных и прогнозирования пространственно-временной структуры популяции в зависимости от возможных изменений климата Земли [13].

СПИСОК ЛИТЕРАТУРЫ

1. *Weiss G.* Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA, USA. 2001. P. 36—61.
2. FIPA Specification [Электронный ресурс]: <http://drogo.cselt.stet.it/fipa/> (по состоянию на 01.02.2007).
3. FIPA-OS: A Component-Based Toolkit Enabling Rapid Development of FIPA Compliant Agents [Электронный ресурс]: <http://fipa-os.sourceforge.net> (по состоянию на 03.04.2006).
4. JADE Programmer's Guide [Электронный ресурс]: <http://jade.tilab.com> (по состоянию на 20.02.2007).
5. JADE Administrator's Guide [Электронный ресурс]: <http://jade.tilab.com> (по состоянию на 20.02.2007).
6. *Collis J., Ndumu D.* Zeus Technical Manual [Электронный ресурс]: <http://labs.bt.com/projects/agents/zeus/techmanual/TOC.html> (по состоянию на 17.04.2006).
7. *Horling B., Lesser V.* The TAEMS White Paper / Univ. of Massachusetts. 2004.
8. *Карпов Ю. Г.* Введение в моделирование с использованием среды AnyLogic [Электронный ресурс]: <http://www.xjtek.com> (по состоянию на 01.12.2006).
9. *Городецкий В. И., Карсаев О. В.* Технология разработки прикладных многоагентных систем в инструментальной среде MASDK // Тр. СПИИРАН. СПб.: Наука, 2006. Вып. 3, т. 1. С. 11—32.
10. ArcGIS Desktop [Электронный ресурс]: <http://esri.com/software/arcgis/about/desktop_gis.html> (по состоянию на 10.04.2008).
11. GeoTools Java Library [Электронный ресурс]: <http://geotools.codehaus.org/> (по состоянию на 10.04.2008).
12. ArcGIS Explorer [Электронный ресурс]: <http://esri.com/software/arcgis/explorer/index.html> (по состоянию на 10.04.2008).
13. *Михайлов В. В., Колпащиков Л. А., Селяков И. С.* Использование агентного подхода к моделированию пространственно-временной динамики северных оленей таймырской популяции // Вопр. природопользования на Крайнем Севере: Сб. науч. тр. СПб.: ГУАП, 2007. С. 38—51.

Сведения об авторах

Владимир Валентинович Михайлов — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: mwwcari@rol.ru

Игорь Сергеевич Селяков — СПИИРАН, лаборатория информационных технологий в системном анализе и моделировании; E-mail: seliakov@mail.ru

Поступила в редакцию
06.05.08 г.

УДК 004.89

С. П. Соколова, Е. А. Кузьмина

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА МОНИТОРИНГА ОСОБО ОПАСНЫХ ДИНАМИЧЕСКИХ ПРОЦЕССОВ

Рассматривается модульная структура интеллектуальной системы мониторинга особо опасных динамических процессов, ориентированной на анализ многомерных первичных данных (точечных и с неопределенностью интервального типа) для исследования динамических свойств и моделирования процессов. Приведен пример результатов мониторинга для процессов в природном очаге чумы.

Ключевые слова: *иммунокомпьютинг, интервальная динамическая система, мониторинг, интеллектуальная система.*

Как известно, в Российской Федерации (территория Кавказа, Прибайкалья, Иркутской области и т.д.) и СНГ (Казахстан, Киргизия, Узбекистан) находятся активные и самые крупные

в мире природные очаги особо опасных динамических процессов (к числу которых можно отнести, в частности, такие заболевания, как сибирская язва, чума и т.д.).

Особо опасные динамические процессы, как правило, являются сложными многокомпонентными динамическими системами. К настоящему времени разработано значительное количество математических моделей и вычислительных процедур для исследования динамики поведения сложных отношений между подсистемами и их компонентами на популяционном, организменном и других уровнях. Однако результаты исследований различных видов математических моделей только дополняют друг друга, что объясняется сложностью таких процессов.

Качественно новый подход к анализу многомерных данных мониторинга особо опасных динамических процессов и распознавания ситуаций был предложен в работах [1—5] на основе интеллектуальной информационной технологии — иммунокомпьютинга. Полученные при этом результаты исследований продемонстрировали высокую вычислительную эффективность при решении задач распознавания стадий особо опасных динамических процессов и прогнозирования их возникновения [1—3], а также возможность формирования и вычисления индексов риска [4, 5].

В настоящей статье представлена модульная структура интеллектуальной системы мониторинга особо опасных динамических процессов, ориентированной на использование точечных и интервальных первичных данных и параметров математических моделей исследуемого объекта. Предложенная интеллектуальная система мониторинга содержит набор модулей и вычислительных процедур, спроектированных на основе традиционных технологий и интеллектуальной информационной технологии — иммунокомпьютинга.

Рассмотрим функциональное назначение модулей интеллектуальной системы мониторинга.

Модуль проектирования реляционной модели данных с временной динамикой. В этом модуле каждое отношение включает время как обязательный атрибут отношения. Реализованная в модуле технология создания базы данных с временной динамикой содержит следующие этапы [3, 6, 7]:

— анализ требований пользователей;

— построение концептуальной, логической и физической моделей базы данных;

— проектирование пользовательских приложений и интерфейса для интеллектуальной системы мониторинга.

Для представления динамики данных в этом модуле формализована временная логика в виде ANU-исчисления [6, 7]. Структура баз данных проектировалась на основе полного перечня всех первичных данных. На базе информации, хранящейся в созданных базах данных с временной динамикой, формируются временные ряды (точечные и интервальные), позволяющие проводить информационный анализ соответствующих индикаторов мониторинга. Эти временные ряды используются в последующем модуле.

Интеллектуальный модуль системы мониторинга. В данном модуле реализованы математические модели и вычислительные процедуры иммунокомпьютинга [1—5, 8—10]. Модуль предназначен для:

— сингулярного разложения плоских и трехмерных матриц с точечными и интервальными элементами [8—11];

— решения задач обучения по тестовым выборкам;

— самообучения системы;

— распознавания стадий особо опасных динамических процессов;

— формирования и вычисления точечных и интервальных значений индексов риска [7].

Выходной информацией модуля являются сформированные обучающие выборки, данные о состояниях особо опасного динамического процесса и вычисленные значения

индексов риска. Эта информация используется в последующем модуле для решения задач идентификации.

Модуль структурной и параметрической идентификации. В этом модуле приведено обоснование выбора структуры математической модели и решение задачи ее параметрической идентификации. При этом были использованы две концепции математического представления моделей:

1) концепция „вход — выход“ — с использованием детерминированных (стохастических) рядов Вольтерры или полиномов Габора — Колмогорова; при таком представлении использовалась аппроксимация экспериментальных кривых методами генетического программирования [7, 12];

2) концепция „пространство состояний“ — в виде дифференциальных или разностных линейных и нелинейных уравнений с запаздыванием или без, с точечными или интервальными параметрами; для нелинейных структур интервальных моделей стадий особо опасного динамического процесса решение задачи параметрической идентификации осуществлялось интервальными методами глобальной безусловной оптимизации [12, 13].

Выходная информация модуля представлена выбранной структурой и восстановленными параметрами математической модели, которые используются в модуле исследования динамических свойств.

Модуль исследования динамических свойств. Этот модуль позволяет осуществлять исследование динамических свойств (управляемость, асимптотическая и экспоненциальная устойчивость, робастность и т. д.) — на основе прямого метода Ляпунова с функцией либо функционалом Ляпунова [12] — следующих типов математических моделей с точечными и интервальными параметрами:

— линейной с наличием запаздывания и без запаздывания;

— нелинейной с нелинейностями секторного типа и квадратичной.

Для математических моделей с интервальными параметрами использованы условия невырожденности интервальной матрицы, непустоты допустимого множества решений интервального матричного уравнения Ляпунова, полученные на основе метода идентифицирующего функционала [12, 13].

Результаты исследований рассмотренных модулей интеллектуальной системы использовались в модуле моделирования динамики распространения особо опасного динамического процесса по ареалу ландшафтно-экологического района природного очага.

Пример. Результаты мониторинга с помощью интеллектуальной системы рассмотрим на примере особо опасной инфекции — чумы в природном очаге. Как известно, в природных очагах чумы периодически возникают эпизоотические процессы (заболевания животных) и даже эпидемиологические ситуации [14, 15]. Вследствие того, что многие природные очаги располагаются в районах залегания природных ресурсов (таких, как нефть, газ и т.д.), их разработка транснациональными компаниями может приводить к выносу и распространению опасных заболеваний далеко за пределы этих районов.

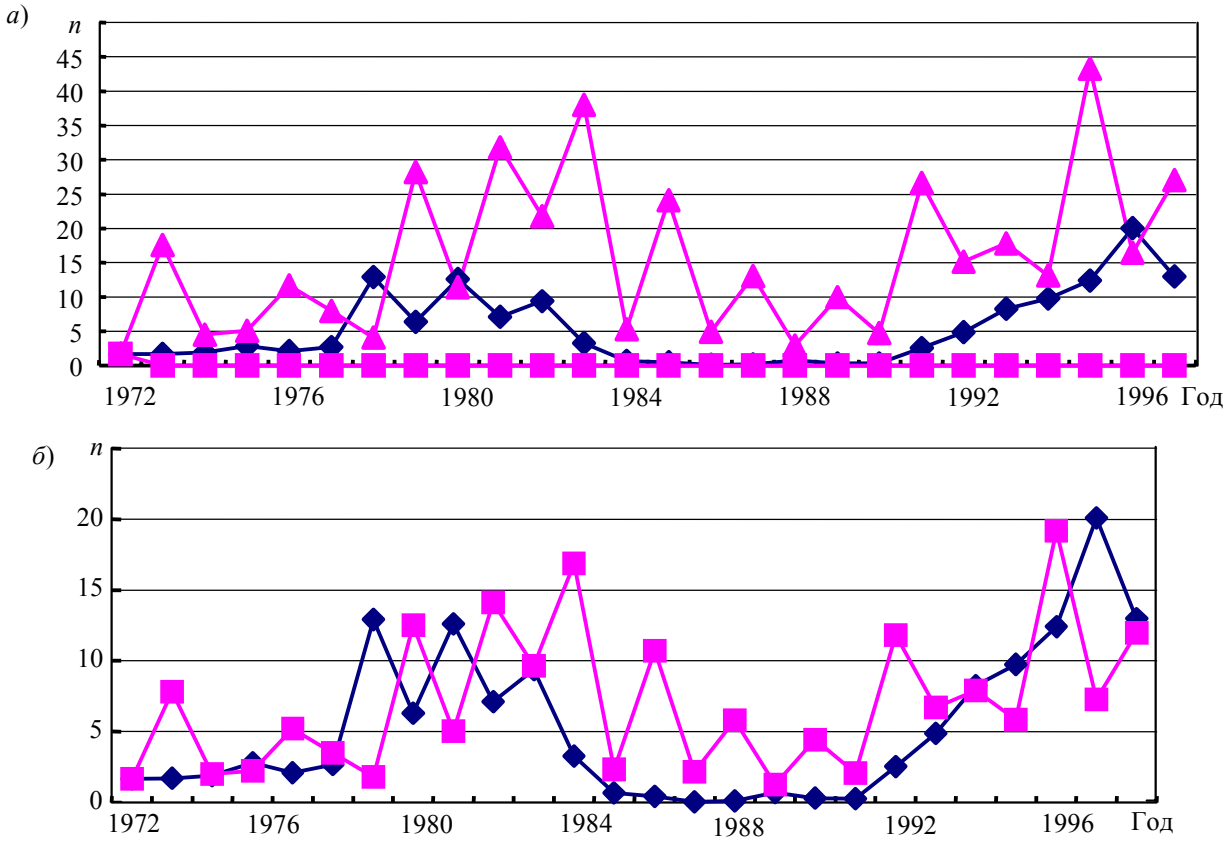
Результаты исследований получены на основе первичных данных по ландшафтно-экологическому району Прибалхашского природного очага чумы (Республика Казахстан) за период с 1949 по 1998 гг., предоставленных исследователями Казахского научного центра карантинных и зоонозных инфекций им. М. Айкимбаева (Алма-Ата) [14, 15].

Чумный эпизоотический процесс является сложной многокомпонентной динамической системой, характеризуемой чумной триадой: носитель — переносчик — чумный микроб. Носителями являются дикие или домашние теплокровные животные (большая песчанка, суслик, верблюд, заяц, кошка и т. д.), переносчиками — различные виды блох, которые обитают на носителях. Возбудитель — чумный микроб — может передаваться посредством прямого контакта, через укус или воздушно-капельным путем.

В интеллектуальной системе мониторинга для представления состояния компонентов чумной триады были использованы наборы индикаторов [14—16]: 8 индикаторов, характеризующих состояние возбудителя, по 12 индикаторов, характеризующих состояния переносчика и носителя, 21 индикатор для оценки влияния внешних (абиотических) факторов.

В качестве примера для демонстрации процесса моделирования динамики численности носителя была выбрана нелинейная интервальная математическая модель, приведенная в работе [12, с. 47, соотношение (3.13)]. С использованием интервального метода глобальной безусловной оптимизации [13] были получены следующие параметры: жесткость внутривидового взаимодействия $\beta = [4,688;\ 5,156]$; интенсивность взаимодействия $\gamma = [19,406;\ 20,00]$; коэффициент, характеризующий скорость размножения популяции носителя при отсутствии переносчика (паразита) и внутривидовой борьбы, $k = [19,703;\ 20,00]$; постоянные коэффициенты: $r = [0,375;\ 0,750]$, $q = [-0,150;\ 0,425]$.

На рисунке представлены результаты моделирования динамики численности (n) носителя при интервальных параметрах модели (средняя квадратическая ошибка $\sigma_1 = 1,933$) и выбранных из вычисленных интервалов средних значениях параметров: $\tilde{\beta} = 4,922$, $\tilde{\gamma} = 19,703$, $\tilde{k} = 19,851$, $\tilde{r} = 0,563$, $\tilde{q} = 0,137$ ($\sigma_2 = 6,271$).



Результаты моделирования динамики численности носителя:
а — при интервальных параметрах модели,
б — при средних (точечных) значениях параметров;
♦ — эксперимент; ▲, ■ — верхняя и нижняя оценки решений

Анализируя представленные на рисунках кривые, можно сделать вывод, что для рассматриваемого случая предпочтительнее использовать математическую модель с интервальными параметрами.

Предложенная интеллектуальная система позволяет эффективно решать задачи мониторинга особо опасных динамических процессов и может быть адаптирована (при наличии первичных данных) для решения подобных задач на любых территориях, включая территории природных очагов особо опасных динамических процессов в Российской Федерации.

Исследования проводились при частичной финансовой поддержке Еврокомиссии (проект МНТЦ К-159-98; проект INCO-COPERNICUS № ICA2-CT-2000-10048).

СПИСОК ЛИТЕРАТУРЫ

1. *Tarakanov A. O., Skormin V. A., Sokolova S. P.* Immunocomputing: Principles and Applications. N.Y.: Springer, 2003.
2. *Tarakanov A., Sokolova S.* et al. Immunocomputing of the natural plague foci // Proc. of the Genetic and Evolutionary Computation Conf. (GECCO-2000), Workshop on Artificial Immune Systems, Las Vegas, USA. 2000. P. 38—41.
3. *Sokolova S. P., Abdullina V. Z.* et al. Artificial Immune System for the Gerbil Natural Plagues Focus / Ed. *A.O. Tarakanov*. Almaty: PC, 2002.
4. *Sokolova L. A.* Index design by immunocomputing // Lecture Notes in Computer Science. 2003. Vol. 2787. P. 120—127.
5. *Соколова Л. А.* Индекс риска чумы на основе иммунокомпьютинга // Тр. СПИИРАН. СПб.: СПИИРАН, 2003. Вып. 1, т. 3. С. 137—141.
6. *Абдуллина В. З.* Проектирование баз данных для противочумной службы Казахстана // Сб. материалов междунар. конф. „Менеджмент и новые технологии“. Алматы, 2001. С. 25—28.
7. *Соколова С. П., Кузьмина Е. А., Абдуллина В. З.* Мониторинг особо опасных инфекций (на примере проблемы чумы) // Математическая биология и биоинформатика, 2007. Т. 2, № 1. С. 82—97. [Электронный ресурс]: <http://www.matbio.org/downloads/Sokolova2007(2_82).pdf>.
8. *Соколова С. П.* и др. Интеллектуальный анализ многомерных данных на основе иммунокомпьютинга. Алматы: РС, 2006.
9. *Кузьмина Е. А.* Градиентный алгоритм сингулярного разложения многомерной интервальной матрицы // Науч. сессия ГУАП: Сб. докл. СПб.: ГУАП, 2007. Ч. 3. С. 148—152.
10. *Соколова С. П., Кузьмина Е. А., Тохтабаев А. Г.* Вычислительная процедура для технического анализа фондового рынка // Тр. СПИИРАН. СПб.: Наука, 2007. Вып. 4. С. 171—183.
11. *Sokolova S. P., Kuzmina E. A., Sokolova L. A.* Analysis and management of a credit risk //Proc. of the 16th Intern. Conf. on Systems Science, Wroclaw. 2007. Vol. 3. P. 375—382.
12. *Sokolova S. P., Ivlev R. S.* Mathematical Modeling and Investigation of Dynamic Properties of Biological Systems at Population Level. Almaty: PC, 2003.
13. *Шарый С. П.* Конечномерный интервальный анализ. Новосибирск: ИВТ СО РАН, 2007.
14. *Айкимбаев А. М.* и др. Эпидемиологический надзор за чумой в Урало-Эмбенском и Предустюртском автономных очагах. Алматы: КазПЧИ, 1994.
15. *Аубакиров С. А.* и др. Руководство по ландшафтно-эпизоотологическому районированию природных очагов чумы Средней Азии и Казахстана. Алма-Ата: КазПЧИ, 1991.
16. *Marshall E. C., Frigessi A., Stenseth N. C.* et al. Plague in Kazakhstan: a Bayesian Model for the Temporal Dynamics of a Vector-Transmitted Infections Disease. Oslo: Univ. of Oslo, 2001.

Сведения об авторах

Светлана Павловна Соколова — СПИИРАН, лаборатория прикладной информатики; E-mail: sokolova_sv@mail.ru

Екатерина Александровна Кузьмина — СПИИРАН, лаборатория прикладной информатики; E-mail: kea@computer.edu.ru

Поступила в редакцию
06.05.08 г.

SUMMARY

P. 7—11.

AGENT-BASED SYSTEM MODELING LANGUAGE

The paper describes base of ASML language that is extension of “standard” UML one used for description of software systems. ASML language is used in instrumental environment MASDK 4.0 for designing of multi-agent systems. Analysis and architectural designing of the systems are executed using graphical diagrams intended for description of systems macro models, interaction protocols, roles’ schemes and behavior scenarios. Detail designing is executed using diagrams intended for description of domain ontology and agent classes’ behavior scenarios.

Keywords: modeling language, multi-agent systems.

Data on authors

Vladimir Ivanovich Gorodetsky — SPIIRAS, Laboratory of Intelligent Systems;
E-mail: gor@iias.spb.su

Oleg Vladislavovich Karsaev — SPIIRAS, Laboratory of Intelligent Systems;
E-mail: ok@iias.spb.su

Vladimir Vladimirovich Samoylov — SPIIRAS, Laboratory of Intelligent Systems;
E-mail: samovl@iias.spb.su

Victor Grigorievich Konyushiy — SPIIRAS, Laboratory of Intelligent Systems;
E-mail: kvg@iias.spb.su

P. 12—16.

AGENT-BASED APPROACH FOR CONFIGURATION OF VIRTUAL ENTERPRISES

An agent-based approach for project scheduling during the formation of virtual enterprises in non-material production is proposed. Each participant of virtual enterprise is represented as independent program agent. We report on our computational results, obtained for the PSPLib benchmark instances.

Keywords: virtual enterprise, distributed project scheduling, agent-based approach.

Data on authors

Victor Grigorievich Konyushiy — SPIIRAS, Laboratory of Intelligent Systems;
E-mail: kvg@iias.spb.su

Oleg Vladislavovich Karsaev — SPIIRAS, Laboratory of Intelligent Systems;
E-mail: ok@iias.spb.su

P. 17—20.

BUILDING COOPERATIVE SELF-ORGANISING NETWORKS: MAJOR TASKS AND TECHNOLOGIES

An approach to solving methodological problems during building cooperative self-organising networks is proposed. The tasks of providing for interoperability between self-organising network members are described together with the principle of selecting standards and protocols for information exchange and negotiation. The proposed approach is based on the idea of knowledge logistics and applies such technologies as ontology management, profiling and intelligent agents.

Keywords: self-organising networks, multi-agent architecture, Web-services, negotiation protocol.

Data on authors

Nikolay Germanovich Shilov — SPIIRAS, Laboratory of Computer-Aided Integrated Systems;
E-mail: nick@iias.spb.su

P. 21—26.

ANALYSIS OF CREDIT BASED MECHANISM OF NETWORK WORM EPIDEMICS DETECTION AND CONTAINMENT

Issues connected with the analysis and updating of credit based mechanisms of detection and containment of network worm epidemics are discussed. The peculiarities of the given protection mechanism, and also a technique and results of its evaluation for various types of network traffic are submitted.

Keywords: network security, network worms, detection and containment of network worm propagation, credit based protection mechanisms, simulation of defense against network worms.

Data on authors

Victor Vasilievich Vorontsov — SPIIRAS, Computer Security Research Group;
E-mail: vorontsov@comsec.spb.ru

Igor Vitalievich Kotenko — SPIIRAS, Computer Security Research Group;
E-mail: ivkote@iias.spb.su

P. 26—30.

SOFTWARE PROTECTION MODEL BASED ON REMOTE ENTRUSTING MECHANISM

An approach based on the remote entrusting to construction of the model of software protection against unauthorized modifications and tampering is suggested. Main composing elements and its functioning principles are considered. Two possible implementations of mobile module replacement based on aspect oriented programming paradigm are proposed.

Keywords: remote entrusting, verification, mobile module, dynamic replacement.

Data on authors

Vasily Alekseevich Desnitsky — SPIIRAS, Computer Security Research Group;
E-mail: desnitsky@comsec.spb.ru

Igor Vitalievich Kotenko — SPIIRAS, Computer Security Research Group;
E-mail: ivkote@iias.spb.su

P. 31—35.

FILTERING POLICY VERIFICATION BASED ON EVENT CALCULUS AND ABDUCTION REASONING

The abductive reasoning approach to verification of filtering policy is considered. The anomaly classification for rules of firewall access control list is proposed. Various scenarios of firewall functioning modeling are analyzed on the base of Event Calculus. Application of abductive search methods for detection and resolution of filtering policy anomalies is presented. These methods are based on disjoint granulation of rule conditions.

Keywords: firewall, traffic filtering, abductive reasoning, network traffic filtering anomaly.

Data on authors

Ekaterina Victorovna Sidelnikova — SPIIRAS, Computer Security Research Group; E-mail: sidelnikova@comsec.spb.ru

Artem Valerievich Tishkov — SPIIRAS, Research Group for Information Technologies in Education; E-mail: avt@iias.spb.su

Igor Vitalievich Kotenko — SPIIRAS, Computer Security Research Group; E-mail: ivkote@iias.spb.su

P. 36—40.

TESTING PROCESS OF FUNCTIONAL MODELS TRANSFORMATION BASED ON HYBRID METHODS

Task of transformation of IDEF0 formal models into UML class diagrams based on hybrid methods including translation rules' ontological design is discussed. Mechanism of transformation rules' creation and equivalence testing of source and derived models is examined.

Keywords: quality control, model testing, model translation, ontological rules.

Data on authors

Alexey Yurievich Podjachev — SPIIRAS, Laboratory of Computer Systems and Problems of Information Protection; E-mail: Alexey.Podjachev@quest.com

Alexey Yurievich Atiskov — SPIIRAS, Laboratory of Computer Systems and Problems of Information Protection; E-mail: atiskov@gmail.ru

Sergey Vladimirovich Perminov — SPIIRAS, Laboratory of Computer Systems and Problems of Information Protection; E-mail: sv.perminov@gmail.com

P. 41—47.

COMPARISON OF METHODS FOR LOCALISATION OF MULTIMODAL SYSTEM USER BY HIS SPEECH

The problem of distant recording and recognition of speech for the task of voice interaction with automatic inquiry system in noisy environment is considered. The system perceives and analyses sounds arising from limited subspace due to spatial localization of sound sources. The test results of three methods for determination of direction to the sound source with using several microphone array schemes are presented.

Keywords: distant speech recognition, microphone array, multimodal interface.

Data on authors

Andrey Leonidovich Ronzhin — SPIIRAS, Laboratory of Speech and Multimodal Interfaces; E-mail: ronzhin@iias.spb.su

Alexey Anatolievich Karpov — SPIIRAS, Laboratory of Speech and Multimodal Interfaces; E-mail: karpov@iias.spb.su

P. 47—51.

AUTOMATIC SYNTACTIC ANALYSIS OF RUSSIAN TEXTS BASED ON THE PHRASE-STRUCTURE GRAMMAR

Conception and implementation of program module of syntactic analysis for the Russian literary language are presented. The main idea of research is based on the phrase-structure grammar, which uses for the formal representation of the syntactic structure, the theoretical explanation of the extraction of the syntactic structures, important for the formal representation of texts in Russian.

Keywords: phrase-structure grammar, syntactic structures, automatic text analysis.

Data on authors

Ildar Amirovich Kagirov — SPIIRAS, Laboratory of Speech and Multimodal Interfaces; E-mail: kagirov@iias.spb.su

Anastasia Borisovna Leontyeva — SPIIRAS, Laboratory of Speech and Multimodal Interfaces; E-mail: an_leo@iias.spb.su

P. 51—56.

CONSIDERING THE PECULIARITY OF SPONTANEOUS SPEECH AT THE DEVELOPING AUTOMATIC SPEECH RECOGNITION SYSTEM

An approach to automatic processing of spontaneous Russian speech, consisting in recognition of non-phonemic elements and modeling of alternative variants of word pronunciation, is considered. A set of acoustic and lexical models, that are intended for separation of noises from keywords and that take into account possible elements of spontaneous speech, is represented. An algorithm of creation of alternative transcriptions by extended transcribing rules is presented. Experimental results are represented.

Keywords: speech recognition, alternative transcriptions, non-phonemic elements.

Data on authors

Alexandra Borisovna Leontyeva — SPIIRAS, Laboratory of Speech and Multimodal Interfaces; E-mail: leonty@iias.spb.su

Irina Sergeevna Kipyatkova — SPIIRAS, Laboratory of Speech and Multimodal Interfaces; E-mail: kipyatkova@iias.spb.su

P. 57—65.

CHOOSING THE MODEL OF FUNCTIONING OF A TECHNICAL SYSTEM FROM THE SET OF ITS ALTERNATIVE MODELS

A theoretical approach to choosing the rules of functioning of a technical system is provided. These rules are selected from the set of alternative models which can be constructed from the initial information about such systems. The proposed approach is based on the formalization of the concepts of the control objective and the goal state, the definition of the rule for unequivocal identification thereof, and the structural features of the system models.

Keywords: model of a technical system, a system state, the control objective.

Data on authors

Nikolay Petrovich Kirillov — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: knp@mail.ru

P. 62—69.

DYNAMIC MODEL OF INTEGRATED PLANNING AND SCHEDULING FOR MODERNIZATION AND FUNCTIONING OF INFORMATION SYSTEM

Original dynamic model of modernization integrated planning, scheduling and functioning of disaster tolerant information system (DTIS), which provides to describe formally a base aspects and peculiarities of concerned interactive processes, is suggested.

Keywords: disaster tolerant information system, integrated planning and scheduling, modernization and functioning of complex objects.

Data on authors

Anna Vladimirovna Ikonnikova — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: ikonnikova@iias.spb.su

Irina Andreevna Petrova — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: irina.petrova.9@gmail.com

Semien Alekseevich Potryasaev — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: semp@mail.ru

Boris Vladimirovich Sokolov — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: sokol@iias.spb.su

P. 69—73.

USING MULTI-AGENT SIMULATOR FOR MODELING DISTRIBUTED SYSTEM

Multi-agent approach for modeling of complex distribution systems is considered. The structure of the multi-agent simulator and agent model, which lay the foundation of it and the principle of interfacing the simulator and geo-information system using Shape-files is explained. The develop program allows to solve any problems for different domain areas, for example — it allow to investigate the spatial-temporal dynamics of the herds and populations of animals with exact geographical positing.

Keywords: multi-agent systems, distribution model, simulator, graphical notion of the data, geo-information systems.

Data on authors

Vladimir Valentinovich Mikhailov — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: mwwcari@rol.ru

Igor Sergeevich Seliakov — SPIIRAS, Laboratory of Information Technologies for Systems Analysis and Modelling; E-mail: seliakov@mail.ru

P. 73—77.

INTELLIGENT MONITORING SYSTEM OF THE ESPECIALLY DANGEROUS PROCESSES

The paper provides a modular architecture of the intelligent monitoring system of the especially dangerous dynamic processes. This system is oriented to multivariate data (pointed and uncertainty interval type) analysis and allows to research dynamic properties and simulate these processes.

This paper includes examples of monitoring results for epizootic processes in the natural plague focus.

Keywords: immunocomputing, interval dynamic system, monitoring, intelligent system.

Data on authors

Svetlana Pavlovna Sokolova — SPIIRAS, Laboratory of Applied Informatics; E-mail: sokolova_sv@mail.ru

Ekaterina Alexandrovna Kuzmina — SPIIRAS, Laboratory of Applied Informatics; E-mail: kea@computer.edu.ru