

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

12(160)
2009

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Загорулько Ю. А., Боровикова О. И. Информационная модель портала научных знаний . . . 2
Савченко В. В., Пономарев Д. А. Оптимизация фонетической базы данных по группе дикторов на основе информационной теории восприятия речи 7

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

- Вашкевич Н. П., Бикташев Р. А. Формализация алгоритма синхронизации процессов при диспетчеризации задач в многопроцессорных системах с использованием механизма "рандеву" 12
Сериков Д. А. К математическому моделированию процесса диспетчеризации задач в распределенной вычислительной среде 17
Коваленко О. Н. Влияние метода обработки очередей на качество обслуживания трафика мультисервисной сети 25
Волосатова Т. М., Беломошцев Д. Е. Технологии и библиотеки методов построения пикосетей мобильных устройств 28

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

- Андреев О. О. Интеграция моделей логического разграничения доступа, описанных на специализированном языке 29
Ушмаев О. С. Проблемы применения биометрической идентификации в комплексной защите информации 34

WEB-ТЕХНОЛОГИИ

- Владыкин А. А., Шалыто А. А. Непроцедурный текстовый язык описания автоматных обработчиков XML-документов и его применение 38
Асратян Р. Э. Служба синхронизации процессов в сети 45

МОДЕЛИРОВАНИЕ

- Оцков Ш. А. Применение модулярной арифметики с фиксированной точкой для ослабления влияния ошибок округления компьютерных вычислений 50
Голубинский А. Н. Методика проверки на адекватность математической модели речевого сигнала экспериментальным данным 54

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

- Анищенко В. С., Булдакова Т. И., Довгалецкий П. Я., Лифшиц В. Б., Гриднев В. И., Суятинов С. И. Концептуальная модель виртуального центра охраны здоровья населения . 59

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

- Махортов С. Д. Интегрированная среда логического программирования LPEexpert. . . . 65
Кирьянов А. А., Сироткин В. Ю., Сироткин Ю. В., Смирнова А. А. Программный комплекс для автоматизированной системы мониторинга экологического состояния на предприятии 67

ДИСКУССИОННЫЙ КЛУБ

- Вяткин В. Б. Синергетический подход к определению количества информации 68
Указатель статей, опубликованных в журнале "Информационные технологии" в 2009 году . . 73
Указатель приложений к журналу "Информационные технологии", опубликованных в 2009 году. 77
Contents 78
Приложение. Валиахметова Ю. И., Мухачева Э. А., Филиппова А. С., Гильманова Н. А., Карипов У. А. Мультиметодная технология ортогональной упаковки и ее применение в задачах транспортной логистики.

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАГИДУЛЛИН Р. Ш.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б. М.
НАРИНЬЯНИ А. С.
НЕЧАЕВ В. В.
ПАВЛОВ В. В.
ПУЗАНКОВ Д. В.
РЯБОВ Г. Г.
СОКОЛОВ Б. В.
СТЕМПКОВСКИЙ А. Л.
УСКОВ В. Л.
ЧЕРМОШЕНЦЕВ С. Ф.
ШИЛОВ В. В.

Редакция:

БЕЗМЕНОВА М. Ю.
ГРИГОРИН-РЯБОВА Е. В.
ЛЫСЕНКО А. В.
ЧУГУНОВА А. В.

Информация о журнале доступна по сети Internet по адресу <http://www.informika.ru/text/magaz/it/> или <http://novtex.ru/IT>.

Журнал включен в систему Российского индекса научного цитирования.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

Ю. А. Загоруйко, канд. техн. наук,
зав. лабораторией,

О. И. Боровикова, мл. науч. сотр.,
Институт систем информатики имени
А. П. Ершова СО РАН, г. Новосибирск,
e-mail: zagor@iis.nsk.su

Информационная модель портала научных знаний

Предложена информационная модель портала научных знаний, который должен обеспечивать содержательный доступ к научным информационным ресурсам. Информационная модель объединяет модели предметной и проблемной областей портала, а также описывает типы представляемой информации. На основе этой модели строятся внутренние хранилища данных портала, организуется его информационное наполнение, навигация и поиск.

Ключевые слова: онтология, портал знаний, информационная модель, научные информационные ресурсы.

Введение

В настоящее время очень остро стоит проблема эффективного использования огромного объема знаний и информационных ресурсов, накопленных в различных областях человеческой деятельности. Однако доступ к этим знаниям и ресурсам существенно ограничен вследствие того, что они слабо структурированы, плохо систематизированы, рассредоточены по различным Интернет-сайтам, библиотекам и архивам. Для решения этой проблемы нами был предложен подход к построению специализированных Интернет-порталов знаний [1, 2], обеспечивающих интеграцию накопленных знаний и информационных ресурсов в определенной области научных знаний и содержательный доступ к ним.

Для обеспечения унифицированного представления разнородных знаний и данных, учета их связанности, а также поддержки функциональности таких порталов предложена информационная модель, базис которой составляет онтология. Описанию этой модели и посвящена данная статья.

1. Информационная модель портала знаний

Информационная модель портала знаний объединяет модели предметной и проблемной облас-

тей, а также описывает типы представляемой на портале информации. Формально информационная модель портала M_p описывается двойкой:

$$M_p = \langle O_p, IC_p \rangle,$$

где O_p — онтология портала; IC_p — информационное содержание (контент) портала.

Онтология является ядром, базовым компонентом информационной модели портала. Она не только описывает систему знаний портала, но и задает формальные структуры для представления его контента.

В данной работе мы используем понятие "онтология" в том смысле, как оно применяется в информатике и искусственном интеллекте [3, 4]. В частности, мы придерживаемся определения онтологии, данного в работе [5], согласно которому онтология является точной спецификацией концептуализации. Здесь под концептуализацией понимается некоторая абстракция, т. е. упрощенное представление мира, построенное для определенной цели. Концептуализация включает объекты, понятия и другие сущности рассматриваемой области, а также отношения между ними.

Для представления онтологии портала необходим формализм, обеспечивающий описание понятий предметной и предметной областей портала и разнообразных семантических связей между ними. Важным требованием к нему является возможность выстраивания понятий предметной области (ПО) в иерархию "общее—частное" и поддержка наследования свойств по этой иерархии. Этот формализм также должен предоставлять возможность задания ограничений на значения свойств объектов ПО и организации логического вывода.

Нами был предложен формализм, удовлетворяющий описанным выше требованиям, который представляет собой метаонтологию вида

$$O = \langle C, R, T, D, A, F, Ax \rangle, \quad (1)$$

где

$C = \{C_1, \dots, C_n\}$ — конечное непустое множество классов, описывающих понятия некоторой предметной или проблемной области;

$R = \{R_1, \dots, R_m\}$, $R_i \subseteq C \times C$, $R = R_T \cup R_P \cup R_A$ — конечное множество бинарных отношений, заданных на классах (понятиях); здесь R_T — антисимметричное, транзитивное, неререфлексивное бинарное отношение наследования, задающее

частичный порядок на множестве понятий C ; R_p — бинарное транзитивное отношение включения ("часть—целое"); R_A — конечное множество ассоциативных отношений;

T — множество стандартных типов;

$D = \{d_1, \dots, d_n\}$ — множество доменов $d_i = \{s_1, \dots, s_k\}$, где s_i — значение стандартного типа $string$;

$TD = T \cup D$ — обобщенный тип данных, включающий множество стандартных типов и множество доменов;

$A = \{a_1, \dots, a_w\}$, $A \subseteq C \times TD \cup R_A \times TD$ — конечное множество атрибутов, т. е. бинарных отношений вида $a_i(C_j, td_k)$ или $a_i(R_j, td_k)$, описывающих свойства понятий C и отношений R_A ;

F — множество ограничений на значения атрибутов понятий и отношений, т. е. предикатов вида $p_i(e_1, \dots, e_m)$, где e_k — это либо имя атрибута ($e_k \in A$), либо константа ($e_k \in td_j$, где $td_j \in TD$);

Ax — множество аксиом, определяющих семантику классов и отношений онтологии.

Отношение R_T служит для задания иерархий на понятиях. Его особенностью является то, что при наследовании от родительского класса его классу-потомку передаются не только все атрибуты, но и отношения.

Отношение включения R_p наделено свойством транзитивности, благодаря этому при поиске объектов (экземпляров классов) можно осуществлять транзитивное замыкание по этому отношению.

Набор ассоциативных отношений R_A определяется потребностями в представлении знаний о связях между объектами в конкретных проблемных и предметных областях. Предложенный формализм позволяет задавать для таких отношений атрибуты, специализирующие связи между аргументами (объектами).

Информационное содержание (контент) портала IC_p строится на основе формальных структур, заданных в метаонтологии O (формула (1)):

$$IC_p = \langle I, R_p, V, A, R_{IC}, R_{IR}, R_{IA}, R_{VTD} \rangle, \quad (2)$$

где

$I = \{i_1, \dots, i_n\}$ — конечное множество экземпляров классов онтологии;

$R_I = \{ri_1, \dots, ri_k\}$ — конечное множество конкретизированных отношений (экземпляров отношений) $ri_i(i_j, i_k)$ между экземплярами классов;

$V = \{v_1, \dots, v_q\}$ — конечное множество конкретных значений обобщенного типа TD ;

$A_I = \{ai_1, \dots, ai_w\}$ — конечное множество конкретизированных атрибутов, т. е. бинарных отношений $ai_i(i_x, v_y)$ или $ai_i(ri_x, v_y)$ между экземпляром класса или отношения и конкретным значением;

$R_{IC} \subseteq I \times C$ — бинарное отношение инцидентности между множеством экземпляров I и множеством классов C ;

$R_{IR} \subseteq R_I \times R$ — бинарное отношение инцидентности между множеством экземпляров отношений R_I и множеством отношений R ;

$R_{IA} \subseteq A_I \times A$ — бинарное отношение инцидентности между множеством конкретизированных атрибутов A_I и множеством A ;

$R_{VTD} \subseteq V \times TD$ — бинарное отношение инцидентности между множеством значений V и обобщенным типом TD .

2. Онтология портала знаний

С содержательной точки зрения онтология портала знаний служит для представления понятий, необходимых для описания как научной деятельности и научного знания в целом, так и конкретной области знаний, в частности.

Для упрощения настройки портала на выбранную область знаний онтология портала разделяется на предметно-независимые (базовые) онтологии и онтологию области знаний портала (предметную онтологию).

В качестве базовых выбраны онтологии научной деятельности O_{RA} и научного знания O_{SK} , которые не зависят от области знаний портала. Эти онтологии строятся на основе формализма метаонтологии O , описанного в разделе 1.

Область знаний портала может включать несколько связанных предметных областей. В связи с этим онтология области знаний портала O_{KD} может включать онтологии нескольких предметных областей O_{SDi} , построенных на основе онтологии научного знания O_{SK} .

Таким образом, онтология портала знаний O_p имеет вид

$$O_p = \langle O_{RA}, O_{KD} \rangle,$$

где O_{RA} — онтология научной деятельности; $O_{KD} = \{O_{SD1}, \dots, O_{SDm}\}$ — онтология области знаний портала.

Онтология научной деятельности [2], выступающая в качестве онтологии проблемной области портала знаний, построена на основе онтологии, предложенной в [6] для описания научно-исследовательских проектов, и фактически является онтологией верхнего уровня. Она включает базовые классы понятий, относящиеся к организации научной и исследовательской деятельности, такие как *Исследователь*, *Организация*, *Событие*, *Деятельность (Проект)*, *Публикация* и др. В эту онтологию также включен класс *Информационный ресурс*, который служит для описания информационных ресурсов, представленных в сети Интернет.

3. Представление данных и информационных ресурсов

Онтология научного знания фиксирует основные содержательные структуры, используемые для построения предметных онтологий. В частности, эта онтология содержит метапонятия, задающие структуры для описания понятий конкретной области знаний, такие как *Раздел науки*, *Метод исследования*, *Объект исследования*, *Предмет исследования*, *Научный результат*.

Понятия каждой предметной онтологии являются реализациями метапонятий онтологии научного знания и могут быть упорядочены в иерархию "общее—частное".

Понятия базовых онтологий связаны между собой ассоциативными отношениями, выбор которых осуществляется не только исходя из полноты представления проблемной и предметной областей портала, но и с учетом удобства навигации по его контенту и поиска информации в нем.

Наиболее важными ассоциативными отношениями являются:

- "научное направление" — связывает события, публикации, организации, исследователей с разделами науки;
- "описывает" — задает связь публикации с научным результатом, объектом или методом исследования;
- "использует" — связывает метод исследования с деятельностью, исследователем или разделом науки;
- "применяется к" — связывает метод исследования с объектом исследования;
- "результат" — связывает научный результат с деятельностью;
- "ресурс" — связывает информационный ресурс с событиями, публикациями, исследователями, методами и объектами исследования;
- "работает в" — связывает исследователя с организацией, в которой он работает.

Заметим, что последнее отношение имеет три дополнительных атрибута *должность*, *дата принятия* и *дата увольнения*, служащих для уточнения, в какой должности и с какого времени работает (работал) исследователь в данной организации.

Аксиомы, включенные в базовые онтологии, позволяют выводить дополнительные ассоциативные отношения между объектами и имеют вид

$$\text{if } r_{p1}(c_1, c_2) \& \dots \& r_{pm}(c_{m-1}, c_m) \text{ then } r_C(c_j, c_k),$$

где $r_{pi} \in R$, $r_C \in R_A$, $c_i \in C$, $i, j, k \in 1 \dots m$.

В качестве примера такой аксиомы приведем аксиому, которая выводит новое ассоциативное отношение "работает в":

$$A1: \text{if } \text{работает_в}(I_1, O_1) \& \& \text{включает}(O_2, O_1) \\ \text{then } \text{работает_в}(I_1, O_2), I_1 \in \text{Исследователь}, \\ O_1, O_2 \in \text{Организация}.$$

Вводя формальные описания понятий области знаний в виде классов объектов и отношений между ними, онтология портала задает структуры для представления реальных объектов и связей между ними. В соответствии с этим данные на портале представлены как множество разнотипных информационных объектов и связей, которые в совокупности образуют информационное содержание или контент портала.

Формально каждый информационный объект (ИО) является элементом множества I , а связи между информационными объектами — элементами класса R_I (см. формулу (2)).

Содержательно ИО является экземпляром какого-либо класса онтологии и представляет описание некоторого объекта моделируемой области знаний или релевантного ей информационного ресурса. Семантика связей, установленных между ИО, определяется отношениями, заданными между соответствующими классами онтологии.

Важным компонентом контента портала является описание информационных ресурсов. Как было сказано выше, каждый ресурс, представленный на портале, соответствует такому понятию онтологии, как *Информационный ресурс*, набор атрибутов и связей которого основан на стандарте Dublin Core [7]. Описание ресурса включает экземпляр понятия *Информационный ресурс* и набор экземпляров отношений, связывающих это понятие с другими понятиями онтологии (*Организациями*, *Исследователями*, *Публикациями*, *Событиями*, *Разделами науки* и т. д.).

4. Организация содержательного доступа

На основе предложенной информационной модели не только строится контент портала знаний, но и организуется содержательный доступ к систематизированным знаниям и информационным ресурсам моделируемой области знаний. Эту функцию обеспечивают развитые средства навигации и поиска.

4.1. Навигация

При навигации по portalу обеспечивается возможность выбора информационных объектов определенного класса, детального просмотра каждого ИО, а также информационных ресурсов, ссылки на которые содержатся в просматриваемом ИО.

Навигация по portalу начинается с выбора некоторого класса в дереве понятий онтологии, построенном на основе отношения R_T . При этом пользователю выдается список информационных

объектов выбранного класса $L(C_X)$, который отображается в виде html-страницы, содержащей набор ссылок на эти объекты. Для больших списков объектов формируется составная страница, включающая список страниц с элементами навигации по этому списку.

Следует заметить, что при формировании списка объектов выбранного класса C_X выполняется транзитивное замыкание по отношению R_T . Вследствие этого результирующий список $L(C_X)$ будет включать как объекты искомого класса C_X , так и объекты его классов-наследников, т. е.

$$L(C_X) = \{i_i | i_i \in C_X \cup C_T\},$$

где $C_X \in C$, $C_T = \{C_Y \in C | \exists R_T(C_X, C_Y)\}$.

Если на информационных объектах класса C_X задано отношение включения R_P , по желанию пользователя множество объектов, включенных в $L(C_X)$, может быть представлено либо в виде списка, либо в виде дерева.

Информация о конкретном объекте и его связях также отображается в виде html-страницы (см. рисунок), формат и наполнение которой зависят от класса данного объекта и заданных для него отношений и шаблона визуализации. При этом объекты, связанные с данным объектом, представляются на его странице в виде гиперссылок, по которым можно перейти к их детальному описанию.

Дальнейшая навигация по portalу представляет собой процесс перехода от одних информационных объектов к другим по заданным между ними связям — экземплярам ассоциативных отношений R_A .

Например, при просмотре информации о конкретном проекте (см. рисунок) мы можем видеть не только значения его атрибутов, но и его связи с другими объектами. Используя представленные связи в качестве элементов навигации, можно перейти к просмотру подробной информации о нем как по прямым отношениям (об объекте исследования, об используемых в проекте методах исследования и полученных в ходе его выполнения научных ре-

зультатах), так и по обратным (об участниках проекта, об описывающих его публикациях и информационных ресурсах).

4.2. Поиск

Поиск информационных объектов базируется на онтологии, благодаря чему пользователю предоставляется возможность задания запроса в терминах области знаний портала.

Поисковые запросы задаются через специальный графический интерфейс, управляемый онтологией портала знаний. При выборе пользователем класса искомого информационных объектов автоматически генерируется поисковая форма, в которой можно задать ограничения на значения атрибутов объектов выбранного класса, а также на значения атрибутов объектов, связанных с данным объектом ассоциативными отношениями.

Проекты	
Название деятельности	AGILE
Описание деятельности	Automatic Generation of Instructions in Languages of Eastern Europe
Дата начала	1 января 1998
Дата окончания	31 декабря 2000
Стадия проекта	завершен
Связи объекта	
Исследует_Объект	
Объекты исследования	Язык
Инструкция	
Результат-Деятельности	
Научные результаты и продукты	
Система AGILE	
Направление деятельности	
Раздел Науки	
Генерация текста	
Понимание текста	
Ссылки на объект	
Персона-Участник-Деятельности	
Персоны	Роль Участника Деятельности
Bateman, J.A.	исполнитель
Hana, J.	исполнитель
Hartley, T.	исполнитель
Kruijff, G.-J.	исполнитель
Kruijff-Korbayová, I.	исполнитель
(Всего: 10)	
Организация-Участник-Деятельности	
Организации	
Information Technology Research Institute, University of Brighton, ITRI	
Institute for Applied Linguistics, University of the Saarland	
Institute of Formal and Applied Linguistics(Charles University), ÚFAL	
Institute of Information Technology, Bulgarian Academy of Sciences, IIT BAS	
РосНИИ искусственного интеллекта, РосНИИ ИИ	
Публикация о Деятельности	
Публикации	
Bateman, J.A., Hana, J., Hartley, T., Kruijff, G.-J., Kruijff-Korbayová, I., Skoumalová, H., Staykova, K., Teich, E., Соколова, Е.Г., Шаров, С.А., A multilingual system for text generation in three slavic languages, 2000, статья	
Bateman, J.A., Kruijff, G.-J., Kruijff-Korbayová, I., Skoumalová, H., Teich, E., Шаров, С.А., Resources for multilingual text generation in three Slavic languages, 2000, статья	
Ресурс-Деятельности	
Интернет-ресурсы	
The AGILE Project (Charles University pages)	
Сайт проекта AGILE	

Представление информационного объекта и его связей

При этом также могут быть заданы условия на значения атрибутов соответствующих отношений.

Формально поисковый запрос, построенный в терминах онтологии, выглядит следующим образом:

$$Q = \{i_X \in C_X | P(i_X) \& Rd(i_X) \& Rb(i_X)\},$$

где

$C_X \in C$ — класс искомых объектов;

$P(i_X) = \big\&_{i=1}^n (a_i(i_X, v_i) \& \big\&_{j=1}^m p_j(v_j, c_j))$ — описание свойств искомого объекта i_X класса C_X , где n — число заданных пользователем атрибутов a_i ; m — число заданных пользователем предикатов p_j на значения v_j атрибутов a_i ; c_j — константа ($c_j \in td_j$, где $td_j \in TD$);

$Rd(i_X) = \big\&_{k=1}^d (ri_k(i_X, i_Y), ri_k \in R_k, i_Y \in C_Y | P(ri_k) \& P(i_Y))$ — описание объектов, состоящих с объектами класса C_X в "прямых" отношениях, где d — число заданных пользователем "прямых" отношений ri_k ; $R_k \in R_P \cup R_A$; $P(ri_k)$ и $P(i_Y)$ — описания свойств экземпляра отношения ri_k и свойств объекта i_Y соответственно, структура которых аналогична структуре описания $P(i_X)$;

$Rb(i_X) = \big\&_{l=1}^b (ri_l(i_Z, i_X), ri_l \in R_l, i_Z \in C_Z | P(ri_l) \& P(i_Z))$ — описание объектов, состоящих с объектами класса C_X в "обратных" отношениях, где b — число заданных пользователем "обратных" отношений ri_l ; $R_l \in R_P \cup R_A$, $P(ri_l)$ и $P(i_Z)$ — описания свойств экземпляра отношения ri_l и свойств объекта i_Z соответственно.

Результатом выполнения запроса Q будет множество объектов искомого класса C_X , удовлетворяющих описанным в запросе свойствам.

Следует заметить, что при выполнении запроса поиск объектов заданного класса C_X может выполняться (по желанию пользователя) с учетом транзитивного замыкания по отношению R_T . В этом случае при обработке запроса будут рассматриваться объекты из множества $\{i_i | i_i \in C_X \cup C_T\}$, где $C_T = \{C_K \in C | \exists R_T(C_X, C_K)\}$.

В качестве результата поиска или при переходе по конкретной связи любого ИО может быть получен достаточно большой список объектов $L(C_X)$ (например, список всех участников крупного проекта или конференции). Для управления такой ситуацией был введен механизм фильтрации, который позволяет сделать выборку из списка $L(C_X)$ путем задания фильтра.

Содержательно фильтр является набором условий, которые определяют допустимые значения

атрибутов информационных объектов и требования к существованию связей с другими ИО. Формально фильтр задается выражением, аналогичным поисковому запросу.

Этот метод позволяет, например, отфильтровать множество участников проекта как по возрасту или научной степени (условия на атрибут), так и по используемым ими методам исследования (условия на связанный объект).

Заключение

В статье рассмотрена информационная модель портала научных знаний, предоставляющего пользователю содержательный доступ к систематизированным знаниям и информационным ресурсам определенной области знаний. Эта модель не только обеспечивает формальное описание системы понятий предметной и проблемной областей портала знаний, но и поддерживает всю его функциональность. На основе этой модели строятся система знаний и внутренние хранилища данных портала, организуются его информационное наполнение, навигация и поиск.

Благодаря использованию онтологий в качестве базиса информационной модели портал знаний представляет собой не просто еще один каталог ресурсов по заданной тематике, а сеть знаний и данных, по связям которой поддерживается удобная навигация и содержательный поиск.

Разделение онтологии портала на предметно-независимые и предметные онтологии делает портал настраиваемым на любую область научных знаний.

На основе предложенной модели совместно с Институтом археологии и этнографии СО РАН был разработан археологический портал знаний [8], обеспечивающий содержательный доступ широкому кругу пользователей к систематизированным знаниям и информационным ресурсам по археологии и этнографии. Позднее этот подход был применен при разработке портала знаний по компьютерной лингвистике [9].

Список литературы

1. Боровикова О. И., Загорюлько Ю. А. Организация порталов знаний на основе онтологий // Компьютерная лингвистика и интеллектуальные технологии. Труды Междунар. семинара "Диалог—2002". (Протвино, 6—11 июня 2002 г.). М.: Наука, 2002. Т. 2. С. 76—82.
2. Загорюлько Ю. А., Боровикова О. И. Технология построения онтологий для порталов научных знаний // Вестник НГУ. Сер. "Информационные технологии". 2007. Т. 5. Вып. 2. С. 42—52.
3. Гаврилова Т. А., Хорошевский В. Ф. Базы знаний интеллектуальных систем: Учебник. СПб.: Питер, 2001. 384 с.
4. Guarino N. Formal Ontology in Information Systems. Proceedings of FOIS'98, Trento, Italy, 6—8 June 1998. Amsterdam: IOS Press. 1998. P. 3—15.

5. Gruber T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing // International Journal of Human-Computer Studies. November 1995. Vol. 43. Issues 5–6. P. 907–928.

6. Benjamins V. R., Fensel D. Community is Knowledge! in (KA)2 / B. R. Gaines, M. A. Musen (eds.) // Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-based Systems workshop, KAW'98. Calgary. SRDG Publications, Department of Computer Science, University of Calgary, 1998.

7. Using Dublin Core. <http://dublincore.org/documents/usageguide/>

8. Андреева О. А., Боровикова О. И., Булгаков С. В., Загорюлько Ю. А., Сидорова Е. А., Циркин Б. Г., Холюшкин Ю. П.

Археологический портал знаний: содержательный доступ к знаниям и информационным ресурсам по археологии // Труды 10-й Национальной конференции по искусственному интеллекту с международным участием (КИИ-2006). М.: Физматлит, 2006. Т. 3. С. 832–840.

9. Боровикова О. И., Загорюлько Ю. А., Загорюлько Г. Б., Кононенко И. С., Соколова Е. Г. Разработка портала знаний по компьютерной лингвистике // Труды 11-й Национальной конференции по искусственному интеллекту с международным участием (КИИ-2008). М.: ЛЕНАНД, 2008. Т. 3. С. 380–388.

УДК 621.372:519.72

В. В. Савченко, д-р техн. наук., проф., зав. каф.,
e-mail: svv@lunn.ru,

Д. А. Пономарев, аспирант, инж.-програм.,
e-mail: dima-tensor@yandex.ru,

Нижегородский государственный
лингвистический университет, г. Н.-Новгород

Оптимизация фонетической базы данных по группе дикторов на основе информационной теории восприятия речи

Ставится и решается задача формирования и оптимизации фонетической базы данных по выбранной группе дикторов. На основе информационной теории восприятия речи предлагается и обосновывается критерий минимума величины информационного рассогласования между объединенной и персональными фонетическими базами данных. Приведены результаты экспериментальных исследований.

Ключевые слова: речь, фонетическая база данных, информационная теория восприятия речи, теоретико-информационный подход.

Введение

Речь, являясь продуктом высшей нервной деятельности человека, играет важнейшую роль в процессе межличностных коммуникаций. Изучение механизмов речеобразования и восприятия является одной из актуальных задач современной теории и практики. С одной стороны, речь каждого человека имеет индивидуальные неповторимые фонетические особенности. Но с другой стороны, указанные особенности не являются преградой в процессе восприятия или понимания речи других людей в рамках одного национального языка, что свидетельствует о наличии определенной общности в фонетическом составе носителей

данного языка. Однако состав любого национального языка не является однородным, так как множество факторов, среди которых ведущую роль занимают географический, культурный и климатический, влияют на формирование определенных языковых групп — диалектов и говоров. Учет данного факта при решении актуальных задач теоретической и прикладной информатики, таких как автоматическое распознавание речи и обучение речи, очевидно, приведет к повышению эффективности методов, используемых при решении указанных задач. Поэтому одним из важнейших вопросов является решение задачи нахождения диктора, имеющего фонетический состав речи максимально полно, но не избыточно описывающий звуковой строй языка определенной общности людей, к которой данный диктор принадлежит. Решению данного вопроса на основе информационной теории восприятия речи (ИТВР) [1] и посвящена настоящая статья.

Постановка задачи

Базовым понятием ИТВР является фонема звукового строя языка, которая определяется через понятие информационного центра-эталона (ИЦЭ) речевого образа [1]. Здесь под речевым образом следует понимать некое множество реализаций одноименных звуковых сигналов $X_r = \{x_r, j\}$, $j = \overline{1, J_r}$, $r = \overline{1, R}$ (J_r — объем r -го множества), а ИЦЭ данного множества $x_r^* \in X_r$ — одна из реализаций, характеризующаяся минимальной суммой информационных рассогласований по Кульбаку—Лейблеру относительно всех других реализаций звуковых сигналов данного речевого образа,

$$x_r^* = \text{Arg} \min_k \sum_{j=1}^{J_r} \rho_{x_r, j}(x_r, k), \quad k = \overline{1, J_r}, \quad (1)$$

где $\rho_{x_r, j}(x_r, k)$ — величина информационного рассогласования (ВИР) между звуками-реализациями x_r, j и x_r, k r -то элемента звукового строя языка.

Известно [2], что ввиду сильной вариативности речи от одного диктора к другому звуковой состав каждого диктора определяется через свое множество ИЦЭ $X_i = \{x_{i,r}^*\}$, $r = \overline{1, R_i}$ — персональную фонетическую базу данных (ПФБД), причем объем базы данных R_i различен для разных дикторов. При этом одноименные фонемы ПФБД разных дикторов образуют группу фонем $\{x_{1,r}^*, x_{2,r}^*, \dots, x_{K_r,r}^*\}$ (K_r — объем r -й группы), характеризующуюся своим ИЦЭ $x_r^{**} = x_{i,r}^*$, $i \in \{1, 2, \dots, K_r\}$. Тогда звуковой строй группы из N дикторов, объединенных по заданному набору их признаков (пол, возраст, регион проживания и т. п.), можно охарактеризовать по средствам объединенной фонетической базы данных (ОФБД) $X = \{x_r^{**}\}$, $r = \overline{1, R_0}$, состоящей из ИЦЭ групп одноименных фонем ПФБД дикторов из группы. Здесь R_0 — это объем ОФБД, и в общем случае $R_0 \neq R_i$, $i = \overline{1, N}$.

Очевидно, что ОФБД малопригодна для описания фонетического строя языка/говора, так как она составлена из ИЦЭ ПФБД разных дикторов, и можно утверждать, что не существует такого носителя языка (диктора), для которого звуковой состав речи описывался бы ОФБД. При этом можно утверждать, что ПФБД разных дикторов имеют разную степень рассогласования с ОФБД. Тогда актуальной становится задача поиска оптимальной ПФБД для группы дикторов, максимально схожей с ОФБД. Именно оптимальная ПФБД больше всего подходит для описания фонетического строя языка, диалекта или говора. Таким образом, поставленная задача разбивается на два этапа:

1) формирование ОФБД для выбранной группы дикторов;

2) поиск оптимальной ПФБД.

Рассмотрим подходы к решению этих задач более подробно.

Синтез алгоритма группирования ПФБД дикторов

Задача формирования ОФБД для группы дикторов по их ПФБД состоит в последовательном объединении одноименных реализаций фонем ПФБД разных дикторов в группы фонем и в последующем определении ИЦЭ по каждой группе фонем согласно формуле (1). Воспользуемся в качестве меры близости различных фонем величиной их информационного рассогласования (ИР или ВИР) в метрике Кульбака—Лейблера [3]:

$$\rho_{x_{r,j}}(x_{r,k}) = \frac{1}{2F} \sum_{f=-F}^F \left| \frac{1 + \sum_{m=1}^P a_{r,j}(m) \exp(-i\pi mf/F)}{1 + \sum_{m=1}^P a_{r,k}(m) \exp(-i\pi mf/F)} \right|^2 - 1, \quad (2)$$

где $a_{r,j}(m)$ и $a_{r,k}(m)$ — авторегрессионные коэффициенты сигналов $x_{r,j}$ и $x_{r,k}$ соответственно, F — верхняя граница частотного диапазона сигнала или используемого для его передачи канала связи, Гц; f — частота, Гц; m — значение счетчика; P — порядок авторегрессионной (АР) модели; i — мнимая единица. Чем ближе в теоретико-информационном смысле расположены сигналы $x_{r,j}$ и $x_{r,k}$, тем меньше значение их ИР $\rho_{x_{r,j}}(x_{r,k})$. Здесь необходимо отметить, что величина $\rho_{x_{r,j}}(x_{r,k})$ не является метрикой или расстоянием в строгом смысле, так как она имеет несимметричный вид относительно сигналов $x_{r,j}$ и $x_{r,k}$. Поэтому воспользуемся в дальнейшем для оценки "расстояния" между сигналами средним значением их ИР, равным $\frac{1}{2} [\rho_{x_{r,j}}(x_{r,k}) + \rho_{x_{r,k}}(x_{r,j})]$. Вводя ограничение "сверху" на ВИР, получим естественное условие группирования фонем: если ВИР между фонемами меньше некоторого порогового уровня

$$\rho_{x_j}(x_k) \leq \rho_0, \quad (3)$$

то фонемы признаются одноименными и объединяются в группу. Это известная формулировка метода обеляющего фильтра [4]. При достаточно большом значении порога в группы будут объединяться разноименные реализации фонем персональных ФБД, в то время как при чрезмерно заниженном значении порогового уровня одноименные реализации фонем будут группироваться в разные группы. На практике значение данного порога следует выбирать из интервала 0,5...1,0.

Тогда алгоритм группирования реализаций фонем ПФБД дикторов группы из N человек в ОФБД состоит в следующей последовательности операций.

1. Персональные ФБД сортируются по возрастанию их объема, и в качестве первого диктора выбирается диктор с минимальным объемом.

2. Первая реализация фонемы ПФБД первого диктора выделяется в отдельную группу фонем ОФБД и становится первой реализацией по группе.

3. Вычисляется ВИР между первой реализацией фонемы первой группы и второй реализацией фонемы ПФБД первого диктора по формуле (2).

4. Если ВИР, найденная в п. 3, не больше порогового уровня ρ_0 , то вторая реализация фонемы ПФБД первого диктора признается одной из реализаций первой группы.

5. Если ВИР, найденная в п. 3, больше порогового уровня ρ_0 , то данная реализация фонемы признается первой реализацией новой группы фонем.

Личностная информация дикторов экспериментальной группы

№№	Фамилия, Имя, Отчество	Пол	Год рождения	Район проживания
1	Аксенова Екатерина Александровна	женский	1989	Южный район
2	Аникина Татьяна Евгеньевна	женский	1985	Центральный район
3	Артамонова Елена Сергеевна	женский	1986	Центральный район
4	Барычева Елена Сергеевна	женский	1988	Северный район
5	Баслина Елена Владимировна	женский	1990	Южный район
6	Беднова Анастасия Павловна	женский	1983	Центральный район
7	Безрукова Дарья Александровна	женский	1991	Южный район
8	Борина Елена Вячеславовна	женский	1988	Северный район
9	Булычева Оксана Александровна	женский	1990	Южный район
10	Великанова Татьяна Александровна	женский	1988	Северный район

6. По формуле (2) вычисляется ВИР между третьей реализацией фонемы ПФБД первого диктора и всеми реализациями фонем всех групп и согласно критерию минимума информационного рассогласования (МИР) [1] определяется реализация фонемы (и группа, которой она принадлежит), для которой ВИР минимальна.

7. Принадлежность третьей реализации фонемы ПФБД первого диктора к найденной в п. 6 группе определяется согласно п.п. 4–5.

8. Пункты 6 и 7 повторяются для всех остальных реализаций фонем ПФБД первого диктора.

9. Пункты 6 и 7 повторяются для всех реализаций фонем ПФБД второго диктора.

М. Пункты 6 и 7 повторяются для всех реализаций фонем ПФБД N -го диктора.

М + 1. Для каждой выделенной группы одноименных фонем определяется ИЦЭ по формуле (1), который и является реализаций фонемы ОФБД группы из N дикторов.

Необходимо отметить, что объемы групп одноименных реализаций фонем K_r , выделенных с помощью алгоритма группирования, неодинаковы для разных групп: большая часть групп состоит из 1 или 2 одноименных реализаций фонем, тогда как существуют группы, у которых объем превышает 10 реализаций фонем. Поэтому вполне логичным является ввод в алгоритм группирования условия отбраковки маргинальных фонем — реализаций фонем, характерных для определенных людей и нехарактерных для изучаемой группы дикторов в целом. Если объем группы не больше заданного порогового уровня, то все реализации фонем данной группы признаются маргинальными и не используются в дальнейших исследованиях:

$$V_r \leq V_0, r = \overline{1, R_0}, \quad (4)$$

где V_r — объем r -й группы одноименных фонем ОФБД; V_0 — пороговый уровень. Использование решающего правила (4) позволяет выделить из ПФБД всех дикторов наиболее общие реализации произношения фонем, характерные для всей группы дикторов в целом. Проиллюстрируем работу предлагаемого алгоритма группирования на следующем примере из практики.

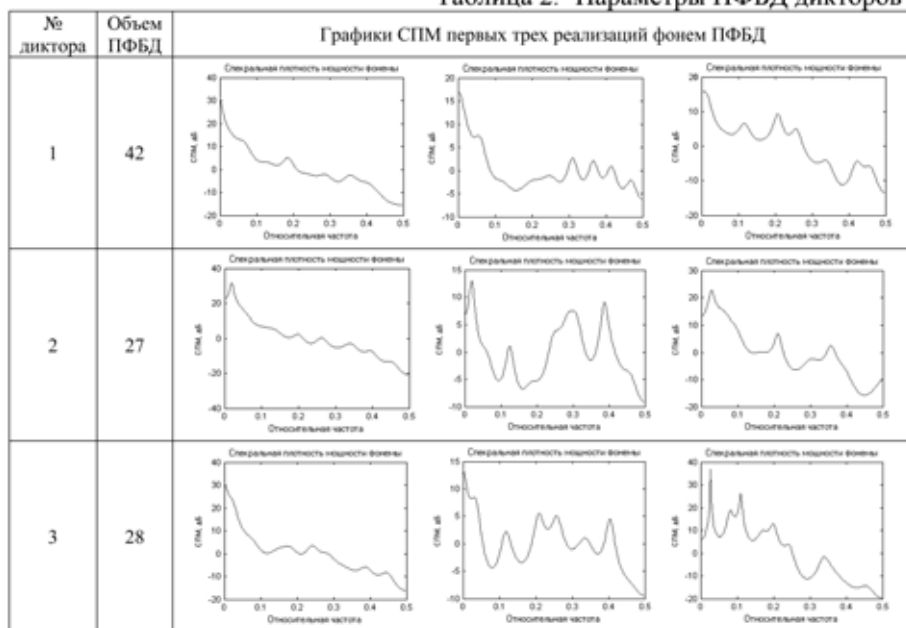
Пример практического применения

Для демонстрации возможностей указанного метода были выбраны 67 дикторов: женщины, в возрасте от 17 до 25 лет, постоянно проживающие на территории Нижегородской области. Фрагмент таблицы с индивидуальными характеристиками первых десяти дикторов экспериментальной группы приведен в табл. 1.

Каждым диктором был проговорен в среднем темпе отрывок объемом около одной страницы из повести А. С. Пушкина "Капитанская дочка". Полученные сигналы сначала были записаны в память ПК в виде соответствующих звуковых файлов. Для этого применялись специальные программные и аппаратные средства — динамический микрофон AKG D77 S и ламповый микрофонный предусилитель ART TUBE MP Project Series USB. Частота дискретизации встроенного АЦП была установлена равной 8 кГц — общепринятая частота при обработке речи. Продолжительность записи по каждому диктору составила около 1,5 мин. Далее, с помощью процедуры автоматического сегментирования речевых сигналов [3] для каждого диктора была составлена своя ПФБД. При этом объем ПФБД разных дикторов варьировался в интервале от 23 до 47 реализаций фонем и составил в среднем по всем дикторам 28 реализаций фонем. Объем ПФБД, а также графики спектральной плотности мощности (СПМ) первых трех реализаций фонем ПФБД для первых трех дикторов из табл. 1 сведены в табл. 2.

На следующем этапе экспериментальных исследований дикторы были разделены на три группы (в зависимости от географии проживания): проживающие в северном, центральном и южном районах Нижегородской области. Для каждой группы из ПФБД дикторов была сформирована ОФБД согласно алгоритму группирования, при этом значение порогового уровня в правиле (3) было выбрано равным 0,6. Далее применялась процедура очистки ОФБД от маргинальных фонем, где значение порога в решающем правиле (4)

Таблица 2. Параметры ПФБД дикторов



равнялось 2. Результаты экспериментальных исследований сведены в табл. 3.

Из табл. 3 видно, что несмотря на разницу в количестве дикторов внутри группы и различных регионах проживания, фонетический состав речи объединенных, очищенных от маргинальных реализаций фонем, фонетических баз данных остается примерно постоянным. Однако, как уже отмечалось выше, ОФБД как искусственная конструкция мало пригодна для описания фонетического состава речи определенной общности людей ввиду наличия в ее составе реализаций фонем разных дикторов. Поэтому актуальной становится задача определения оптимальной ПФБД, имеющей наименьшую степень рассогласования с ОФБД. Для решения поставленной задачи предлагается новый критерий минимума ВИР между ОФБД и ПФБД.

Критерий минимума ВИР

Для определения степени схожести каждой персональной ФБД с объединенной ФБД введем понятие ВИР между ФБД. Для этого рассмотрим две ФБД: $X_1 = \{x_1, r\}$, $r = \overline{1, R_1}$ и $X_2 = \{x_2, r\}$,

$r = \overline{1, R_2}$, причем в общем случае их объемы не равны, т. е. $R_1 \neq R_2$. Построим для данных ФБД матрицу ВИР (табл. 4), где $\rho_{x_1, i}(x_2, j)$ — ВИР между i -й реализацией фонемы ФБД X_1 и j -й реализацией фонемы ФБД X_2 , определяемой по формуле (2). Тогда по каждому столбцу и по каждой строке матрицы ВИР (см. табл. 4) можно определить минимальное значение ВИР, а ВИР между разными ФБД можно определить как среднее значение этих величин:

$$\rho_{X_1}(X_2) = \frac{1}{R_1 + R_2} \left\{ \sum_{j=1}^{R_2} \min_i \rho_{x_1, i}(x_2, j) + \sum_{i=1}^{R_1} \min_j \rho_{x_1, i}(x_2, j) \right\}. \quad (5)$$

Чем меньше значение ВИР $\rho_{X_1}(X_2)$, тем "ближе" в теоретико-информационном смысле находятся ФБД X_1 и X_2 .

Таким образом, задача поиска оптимальной ПФБД сводится к последовательному вычислению ВИР между ОФБД и персональными ФБД, входящими в состав группы, по формуле (5). ПФБД, для которой ВИР относительно ОФБД минимальна, и является оптимальной, а именно:

$$X_{opt} = \text{Arg}_k \min \{ \rho_{X_0}(X_k) \}, k = \overline{1, N}, \quad (6)$$

где X_{opt} — оптимальная ПФБД в группе; X_0 — ОФБД; X_k — ПФБД k -го диктора из группы; N — число дикторов в группе.

По сути, операция поиска минимальных значений ВИР по столбцам матрицы ВИР предлагаемого критерия эквивалента классификации реализаций фонем ПФБД по отношению к ИЦ-эталонам ОФБД или операции определения одноименных групп фонем для рассматриваемых

Таблица 3

Сводная информация об ОФБД

Название группы	Пол	Возраст	Район проживания (для Нижегородской области)	Число дикторов	Объем ОФБД	Число маргинальных групп фонем
Группа № 1	жен	17–25 лет	Северный	12	149	113
Группа № 2	жен	17–25 лет	Центральный	22	293	255
Группа № 3	жен	17–25 лет	Южный	33	429	386

Таблица 4

Матрица ВИР между реализациями фонем ФБД

	$x_{2,1}$	$x_{2,2}$...	x_{2,R_2}
$x_{1,1}$	$\rho_{x_{1,1}}(x_{2,1})$	$\rho_{x_{1,1}}(x_{2,2})$...	$\rho_{x_{1,1}}(x_{2,R_2})$
$x_{1,2}$	$\rho_{x_{1,2}}(x_{2,1})$	$\rho_{x_{1,2}}(x_{2,2})$...	$\rho_{x_{1,2}}(x_{2,R_2})$
...
x_{1,R_1}	$\rho_{x_{1,R_1}}(x_{2,1})$	$\rho_{x_{1,R_1}}(x_{2,2})$...	$\rho_{x_{1,R_1}}(x_{2,R_2})$

Таблица 5

Фрагмент матрицы ВИР между ИЦЭ ОФБД и реализациями фонем ПФБД первого диктора из группы № 1

	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$
$x_{1,1}$	0,2013	0,1018	0,0008	0,4136	0,8011	0,0018
$x_{1,2}$	1,0024	2,2033	0,0005	2,3011	0,5405	1,0634
$x_{1,3}$	0,0142	0	5,0023	2,0110	0,9850	0,0021
$x_{1,4}$	0,0071	0,0034	0	0,6980	6,0729	0,0445
$x_{1,5}$	2,0033	3,0054	1,0037	0,8151	4,0011	2,4420

ФБД. Тогда величина $\min_i \rho_{x_{0,i}}(x_{k,j})$ определяет меру "близости" между одноименными реализациями фонем ОФБД и ПФБД. Чем она меньше, тем "ближе" в теоретико-информационном смысле одноименные реализации фонем рассматриваемых ФБД. Эта же логика рассуждений справедлива и для операции поиска минимальных значений ВИР по строкам матрицы ВИР. Следовательно, среднее значение ВИР между одноименными реализациями фонем ФБД характеризует меру "близости" самих ФБД по отношению друг к другу, и чем значение ВИР (5) меньше, тем "ближе" ФБД. Таким образом, предлагаемый критерий минимума ВИР позволяет определить оптимальную ПФБД из группы как максимально схожую с ОФБД.

Для иллюстрации возможностей данного критерия осуществим поиск оптимальной ПФБД для группы № 1, выделенной в предыдущем примере. Для этого согласно описанному выше критерию осуществлялся расчет матрицы ВИР между ИЦЭ ОФБД и реализациями фонем персональных ФБД. При этом ВИР определялась по формуле (2), а для расчета авторегрессионных коэффициентов использовалась высокоскоростная рекуррентная процедура Берга—Левинсона [5]:

$$a_m(i) = a_{m-1}(i) + c_m a_{m-1}(m-i), \quad i = \overline{1, m};$$

$$c_m = -2 \frac{\sum_{t=m+1}^N \eta_{m-1}(t) v_{m-1}(t-1)}{\sum_{t=m+1}^N [\eta_{m-1}^2(t) + v_{m-1}^2(t-1)]};$$

$$\eta_m(t) = \eta_{m-1}(t) + c_m v_{m-1}(t-1); \quad (7)$$

$$v_m(t) = v_{m-1}(t-1) + c_m \eta_{m-1}(t), \quad t = \overline{1, M};$$

$$\sigma_m^2 = (1 - c_m^2) \sigma_{m-1}^2,$$

$$\sigma_0^2 = \frac{1}{N} \sum_{t=1}^N |x(t)|^2, \quad m = \overline{1, P},$$

при ее инициализации системой равенств $v_0(t) = \eta_0(t-1) = x(t)$. Здесь a_m — авторегрессионные коэффициенты порядка m ; c_m — коэффициент отражения; $\eta_m(t)$ — ошибка линейного предсказания вперед; $v_m(t)$ — ошибка линейного предсказания назад; σ_m^2 — дисперсия возбуждающего белого шума; P — порядок АР-модели; M — объем выборки. Фрагмент матрицы ВИР между ИЦЭ ОФБД и реализациями фонем ПФБД первого диктора группы № 1 приведен в табл. 5.

Далее по каждой строке и по каждому столбцу находились минимальные значения ВИР и по формуле (5) вычислялось значение ВИР между ОФБД и ПФБД. Описанная процедура повторялась для ПФБД всех дикторов, входящих в группу № 1, как в режиме очистки от маргинальных фонем, так и без него. Результаты экспериментальных исследований приведены на рис. 1 и 2 соответственно. Из рис. 1 видно, что минимум зависимости значения ВИР между ОФБД и ПФБД группы дикторов соответствует диктору под номером 9 — Мартынова Жанна Владимировна. Однако использование режима очистки ФБД от маргинальных фонем (4) существенно меняет характер кривой (рис. 2) и положение минимума (диктор № 2 — Борина Елена Вячеславовна). Следовательно, ПФБД диктора № 9 содержит большое число маргинальных, свойственных только данному диктору, реализаций фонем, тогда как ПФБД диктора № 2 наиболее близка по звуковому составу к ОФБД, лишена большого числа мар-

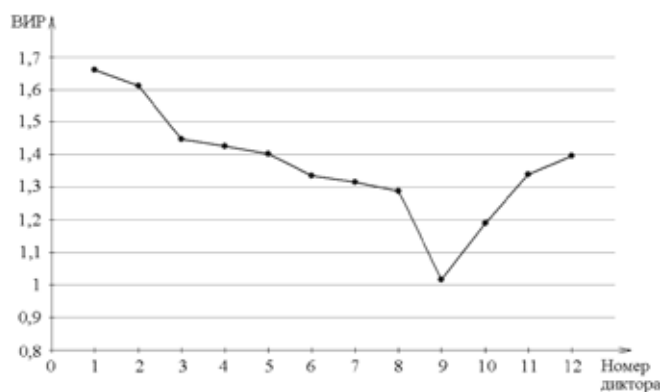


Рис. 1. ВИР между ОФБД и ПФБД группы дикторов

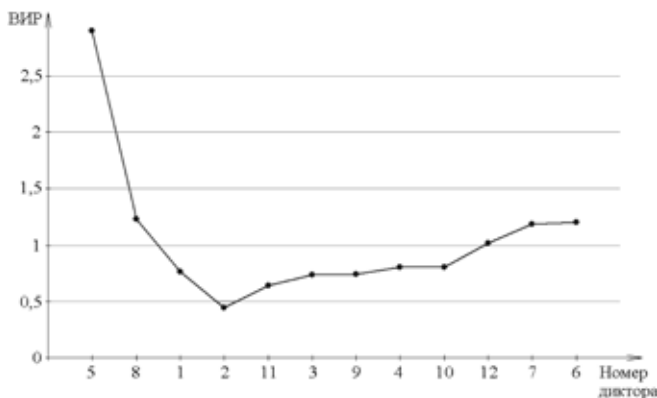


Рис. 2. ВИР между очищенной от маргинальных фонем ОФБД и ПФБД группы дикторов

гинальных фонем и может быть признана оптимальной согласно критерию (б).

Таким образом, благодаря проведенному исследованию предложен новый метод оптимизации фонетической базы данных на основе критерия минимума величины информационного рассогласования между объединенной и персональными фонетическими базами данных. Его главное звено — это анализ ФБД с точки зрения определения оптимальной персональной фонетической базы данных, вносящий максимум полез-

ной информации в объединенную фонетическую базу данных. Сначала из персональных ФБД группы дикторов, выбранных по заданным критериям отбора, строится ОФБД, затем проводится очистка ФБД от маргинальных фонем и далее согласно предложенному критерию определяется степень близости персональных ФБД каждого диктора с объединенной ФБД. Оптимальной признается ПФБД такого диктора из группы, значение величины информационного рассогласования с объединенной ФБД которого минимально. Именно оптимальная ПФБД больше всего подходит для описания фонетического строя выбранной группы или общности людей.

Список литературы

1. Савченко В. В. Информационная теория восприятия речи // Известия вузов. Радиоэлектроника. 2007. Вып. 6. С. 3—9.
2. Савченко В. В., Губочкин И. В. Фонетический анализ речи методом переменного дерева // Известия вузов. Радиоэлектроника. 2008. Вып. 3. С. 14—20.
3. Савченко В. В., Пономарев Д. А. Автоматическое сегментирование речевых сигналов методом обеляющего фильтра. // Известия вузов. Радиоэлектроника. 2008. Вып. 4. С. 18—25.
4. Савченко В. В. Автоматическая обработка речи по критерию минимума информационного рассогласования на основе метода обеляющего фильтра // Радиотехника и электроника. 2005. Т. 50. № 3. С. 309—314.
5. Марпл С. Л. Цифровой спектральный анализ и его приложения. М.: Мир, 1990. 584 с.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ И СЕТИ

УДК 004.3.012

Н. П. Вашкевич, д-р техн. наук, проф., Пензенский государственный университет,

Р. А. Бикташев, канд. техн. наук, проф., Пензенская государственная технологическая академия,
e-mail: bir@pgta.ac.ru

Формализация алгоритма синхронизации процессов при диспетчеризации задач в многопроцессорных системах с использованием механизма "рандеву"

Рассматриваются вопросы формального представления алгоритмов синхронизации взаимодействующих процессов, действующих в параллельных вычислительных системах, при выполнении диспетчеризации задач, находящихся в очереди к процессорам типа FIFO. В алгоритмах синхронизации применяются механизм "рандеву" и примитив "монитор". Для формализации использовалась логика недетерминированных автоматов. Получены уравнения, описывающие процедуры клиентской и серверной частей алгоритма синхронизации с учетом принятой дисциплины обеспечения приоритетности и взаимного исключения процессов при выборе исполнителей задач (процессоров).

Ключевые слова: многопроцессорные системы, диспетчеризация задач, синхронизация процессов, недетерминированный автомат, алгоритм "работа парикмахерской", механизм "рандеву".

Введение

При проектировании многопроцессорных операционных систем остро стоит проблема уменьшения накладных расходов, возникающая, в ча-

стности, при планировании процессов. Частью планировщика является функция диспетчеризации задач (процессов, потоков) при их назначении процессорным узлам. Реализация этой функ-

ции тесно связана с необходимостью синхронизации взаимодействующих процессов. Традиционно в однопроцессорных системах синхронизация процессов осуществляется программным путем в пространстве ядра операционной системы или в пространстве пользователя и на производительность вычислительной системы коренным образом не влияет. В многопроцессорных системах относительные временные затраты на синхронизацию процессов увеличиваются. С достаточно высокой точностью можно допустить, что для одной и той же программы, выполняющейся в однопроцессорном и многопроцессорном режимах, временные затраты на синхронизацию процессов одинаковы. Относительные же временные затраты резко различаются вследствие уменьшения времени выполнения параллельных потоков. Это объясняется тем, что та часть программы, которая связана с синхронизацией процессов, является последовательной и, в соответствии с законом Амдала [1], является фактором, снижающим производительность многопроцессорной системы.

Рассмотрим многопроцессорную систему, в которой используется алгоритм планирования с разделением загрузки [2]. Метод заключается в реализации глобальной очереди задач типа FIFO, разделяемой между всеми процессорами. Вновь поступившая задача помещается в конец очереди, причем задачи, находящиеся в начале очереди, являются кандидатами на выполнение. Когда один из процессоров становится свободным, он выбирает из головы очереди готовую к выполнению задачу и работает с ней до ее завершения или до момента блокирования, например, вследствие необходимости выполнения операции ввода-вывода.

Алгоритм разделения загрузки представляет собой наиболее простой и одновременно эффективный способ планирования, поскольку обладает рядом достоинств, а именно:

- загрузка распределяется равномерно между процессорами, обеспечивая отсутствие простоев процессоров при наличии готовых к выполнению задач;
- простота представления и высокая степень понятности алгоритма функционирования планировщика, заключающаяся в том, что когда процессор освобождается, он вызывает функцию назначения задач (диспетчера) из операционной системы.

Эти достоинства обеспечивают преимущественное применение такого способа планирования в многопроцессорных системах. Однако его практическая программная реализация в пространстве пользователя, хотя и является быстрой, однако вызывает затруднения, поскольку для выполнения сопряженной с диспетчеризацией процедуры синхронизации требуются три семафора:

- счетчик — для подсчета числа мест, занятых готовыми для выполнения задачами;
- счетчик — для подсчета числа процессоров, занятых обслуживанием;
- мьютекс — для реализации взаимного исключения, предотвращающего одновременный доступ нескольких свободных процессоров к единственной очереди, которая в данном случае играет роль общего ресурса.

Очевидным выходом из этого положения является реализация процедуры синхронизации в пространстве ядра, однако возрастающие при этом временные затраты резко уменьшают производительность многопроцессорной системы.

Одним из путей решения проблемы является аппаратная реализация функций синхронизации, которая в значительной степени снимает проблему временных потерь. Кроме того, этим достигается увеличение надежности операционной системы, что особенно важно для проблемно-ориентированных и специализированных систем.

Аппаратная реализация требует поиска методов формализации алгоритмов функционирования, способных упростить и облегчить процесс проектирования соответствующих устройств. В данной работе предлагается аппарат недетерминированных автоматов для решения поставленной задачи [3].

Формализация алгоритма синхронизации

В самом общем виде алгоритм диспетчеризации задач выглядит следующим образом. В начальном состоянии каждый включающийся в работу процессор выполняет операцию "готов" и переходит в спящий режим, в котором находится до тех пор, пока на обслуживание не поступит задача. Если все процессоры заняты обслуживанием, то вновь поступившая задача помещается в конец очереди, число мест в которой ограничено. Если вновь поступившая задача обнаружит, что очередь заполнена, она на обслуживание не принимается и покидает систему. Принятая на обслуживание задача находится в очереди до тех пор, пока не поступит на выполнение в процессор, при этом в очереди освобождается одно место. После выполнения очередной задачи процессор просматривает очередь, и если в ней имеются ожидающие задачи, то он берет на выполнение задачу, стоящую в голове списка. Если очередь пуста, процессор переходит в режим ожидания. Если освободилось одновременно несколько процессоров, то осуществляется выбор одного из них для обслуживания очередной задачи.

В общем случае алгоритм диспетчеризации в явном виде связан с взаимодействием процессов: с одной стороны — задачи (процессы, потоки), тре-

бующие своего выполнения, с другой стороны — обслуживающие их процессорные узлы. Эти действия необходимо синхронизировать таким образом, чтобы обеспечить так называемое "рандеву" [4], когда j -й процессорный узел должен дожидаться поступления i -й задачи, а i -я задача — освобождения одного из процессоров, после чего она будет обслуживаться в течение некоторого времени.

Следует учесть обстоятельство, что при одновременном освобождении нескольких процессоров они одновременно обращаются к общей очереди задач, вызывая тем самым конфликт, который разрешается методом взаимных исключений с использованием приоритетов задач.

Традиционный подход, связанный с таким обслуживанием, аналогичен задаче "спящего парикмахера" [5, 6], который широко применялся для вычислительных систем с одним процессором. Для многопроцессорных систем такой подход должен быть расширен до задачи, которую можно назвать "работа парикмахерской". Эта задача иллюстрирует отношения "клиент—сервер", которые имеют место между процессами в многопроцессорных вычислительных системах, причем аналогом процессоров является коллектив парикмахеров, а аналогом задач выступают клиенты парикмахерской.

Суть задачи о работе парикмахерской заключается в следующем. В одном из городков есть парикмахерская с одной входной и N выходными дверями, в которой размещены m кресел для посетителей и N кресел парикмахеров, в которых они обычно сидят и спят, если в салоне отсутствуют клиенты. В кресла парикмахеров садятся и клиенты, когда происходит процедура стрижки.

Когда посетитель приходит и видит спящих парикмахеров, то он выбирает одного из них, будит его и, подождя, когда он освободит кресло, садится в него. После этого парикмахер может приступить к стрижке посетителя. Если все парикмахеры заняты стрижкой на момент прихода нового посетителя, то последний или садится в свободное клиентское кресло и засыпает, или уходит, если нет свободных кресел.

После стрижки парикмахер открывает посетителю выходную дверь и закрывает ее за ним. В том случае, если в салоне есть ожидающие посетители, то освободившийся парикмахер будит одного из них и ждет, пока тот сядет в его кресло. Если посетителей нет, то парикмахер садится в свое кресло и спит до прихода следующего посетителя.

Рассматривая посетителей и парикмахеров как взаимодействующие процессы в системе, для которых посетитель — это клиент, запрашивающий сервис у парикмахера, а парикмахер — это сервер, обеспечивающий данный сервис, можно предста-

вить данный тип взаимодействия как пример отношения "клиент—сервер".

Действия парикмахеров и посетителей необходимо синхронизировать таким образом, чтобы обеспечить j -му парикмахеру и i -му посетителю встречу — рандеву, когда парикмахер должен дожидаться прихода посетителя, а посетитель — освобождения парикмахера. При этом посетителю необходимо ждать, пока парикмахер закончит его стричь, после чего он освобождает кресло парикмахера и ждет открытия двери парикмахером. Парикмахер, в свою очередь, перед тем как закрыть дверь, должен подождать, пока уйдет посетитель. Таким образом, парикмахер и посетитель проходят через последовательность синхронизированных этапов, начинающихся с рандеву.

Для формального описания алгоритма взаимодействия процессов в данной задаче будем базироваться на использовании моделей недетерминированных автоматов и их представления в виде систем канонических уравнений, описывающих все реализуемые события управляющего алгоритма [1]. Для этой цели введем следующие основные частные события:

$S_{ОД}^К$ и S_K^K — события, свидетельствующие о том, что клиент открыл входную дверь и что в салоне имеется свободное клиентское кресло соответственно;

S_0^K и $S_0^П$ — события, определяющие нахождение клиентов и парикмахеров в салоне парикмахерской соответственно;

$S_{БПj}^К$ и $S_{БК}^{Пj}$ — события, свидетельствующие о том, что очередной клиент будит j -го парикмахера, а j -й парикмахер будит клиента соответственно;

$S_{КПj}^К$ и $S_{ОК}^{Пj}$ — события, свидетельствующие о том, что очередной клиент сел в кресло j -го парикмахера, а j -й парикмахер освободил свое кресло соответственно;

S_C^K и $S_C^{Пj}$ — события, свидетельствующие о том, что очередной клиент и j -й парикмахер спят соответственно;

$S_C^П$ — событие, свидетельствующее о том, что в салоне имеются спящие парикмахеры;

S_g^K и $S_g^{Пj}$ — события, свидетельствующие о том, что очередной клиент и j -й парикмахер готовы к обслуживанию соответственно;

$S_P^{Пj}$ — событие, свидетельствующее о том, что выбран j -й парикмахер для обслуживания клиента;

$S_{СТ}^{Пj}$ — событие, свидетельствующее о том, что j -й парикмахер выполняет стрижку;

$S_{ЗР}^{Пj}$ — событие, свидетельствующее о том, что j -й парикмахер закончил стрижку;

$S_{ОК}^К$ и $S_{ОД}^{Пj}$ — события, свидетельствующие о том, что очередной клиент освободил кресло

j -го парикмахера, а j -й парикмахер открыл выходную дверь соответственно;

$S_{\text{У}}^{\text{К}}$ и $S_{\text{ЗД}}^{\text{П}j}$ — события, свидетельствующие о том, что обслуженный клиент ушел из парикмахерской, а j -й парикмахер закрыл за ним выходную дверь соответственно.

На основании словесно представленного алгоритма управления процессами в задаче о работе парикмахерской и введенных событий, реализуемых в этом алгоритме, система канонических уравнений, описывающих эти события, будет иметь следующий вид:

- для процесса "клиент":

$$S_0^{\text{К}}(t+1) = S_{\text{ОД}}^{\text{К}} S_{\text{К}}^{\text{К}};$$

$$S_{\text{БП}j}^{\text{К}}(t+1) = S_0^{\text{К}} S_{\text{С}}^{\text{П}} S_{\text{Р}}^{\text{П}j};$$

$$S_{\text{КП}j}^{\text{К}}(t+1) = S_{\text{БП}j}^{\text{К}} S_{\text{ОК}}^{\text{К}} \vee S_{\text{С}}^{\text{К}} S_{\text{Р}}^{\text{П}j} S_{\text{ОК}}^{\text{К}} S_{\text{БК}}^{\text{П}j};$$

$$S_g^{\text{К}}(t+1) = S_{\text{КП}j}^{\text{К}} S_{\text{ЗД}}^{\text{П}j} \vee S_g^{\text{К}} \bar{S}_g^{\text{П}j};$$

$$S_{\text{С}}^{\text{К}}(t+1) = S_0^{\text{К}} \bar{S}_{\text{С}}^{\text{П}} \vee S_{\text{С}}^{\text{К}} S_{\text{Р}}^{\text{П}j} \bar{S}_{\text{БК}}^{\text{П}j};$$

- для процесса "сервер":

$$S_0^{\text{П}} = \bigvee_{i=1, \bar{N}} S_0^{\text{П}i};$$

$$S_{\text{БК}}^{\text{П}j}(t+1) = S_0^{\text{П}j} S_{\text{О}}^{\text{К}} S_{\text{Р}}^{\text{П}j};$$

$$S_{\text{С}}^{\text{П}j}(t+1) = S_0^{\text{П}j} \bar{S}_{\text{О}}^{\text{К}} \vee S_{\text{С}}^{\text{П}j} S_{\text{Р}}^{\text{П}j} \bar{S}_{\text{БП}}^{\text{К}};$$

$$S_{\text{ОК}}^{\text{П}j}(t+1) = S_{\text{С}}^{\text{П}j} S_{\text{Р}}^{\text{П}j} S_{\text{БП}}^{\text{К}};$$

$$S_g^{\text{П}j}(t+1) = (S_{\text{БК}}^{\text{П}j} \vee S_{\text{ОК}}^{\text{П}j}) S_{\text{КП}j}^{\text{К}} \vee S_g^{\text{П}j} \bar{S}_g^{\text{К}}.$$

Система канонических уравнений, описывающая события после randevu:

$$S_{\text{СТ}}^{\text{П}j}(t+1) = S_g^{\text{К}} S_g^{\text{П}j} \vee S_{\text{ЗР}}^{\text{П}j} \bar{S}_{\text{СТ}}^{\text{П}j};$$

$$S_{\text{ОК}}^{\text{К}}(t+1) = S_{\text{СТ}}^{\text{П}j};$$

$$S_{\text{ОД}}^{\text{П}j}(t+1) = S_{\text{СТ}}^{\text{П}j};$$

$$S_{\text{ЗД}}^{\text{П}j}(t+1) = S_{\text{ОД}}^{\text{П}j} S_{\text{У}}^{\text{П}j};$$

$$S_{\text{У}}^{\text{К}}(t+1) = S_{\text{ОК}}^{\text{К}} S_{\text{ОД}}^{\text{П}j}.$$

Уравнениям соответствует граф недетерминированного автомата (НДА), представленный по частям на рис. 1, 2 и 3. На рис.1 и 2 представлены части графа клиентской и серверной задач "работа парикмахерской" до момента "randevu", а на рис. 3 — часть графа задачи "работа парикмахерской" после "randevu".

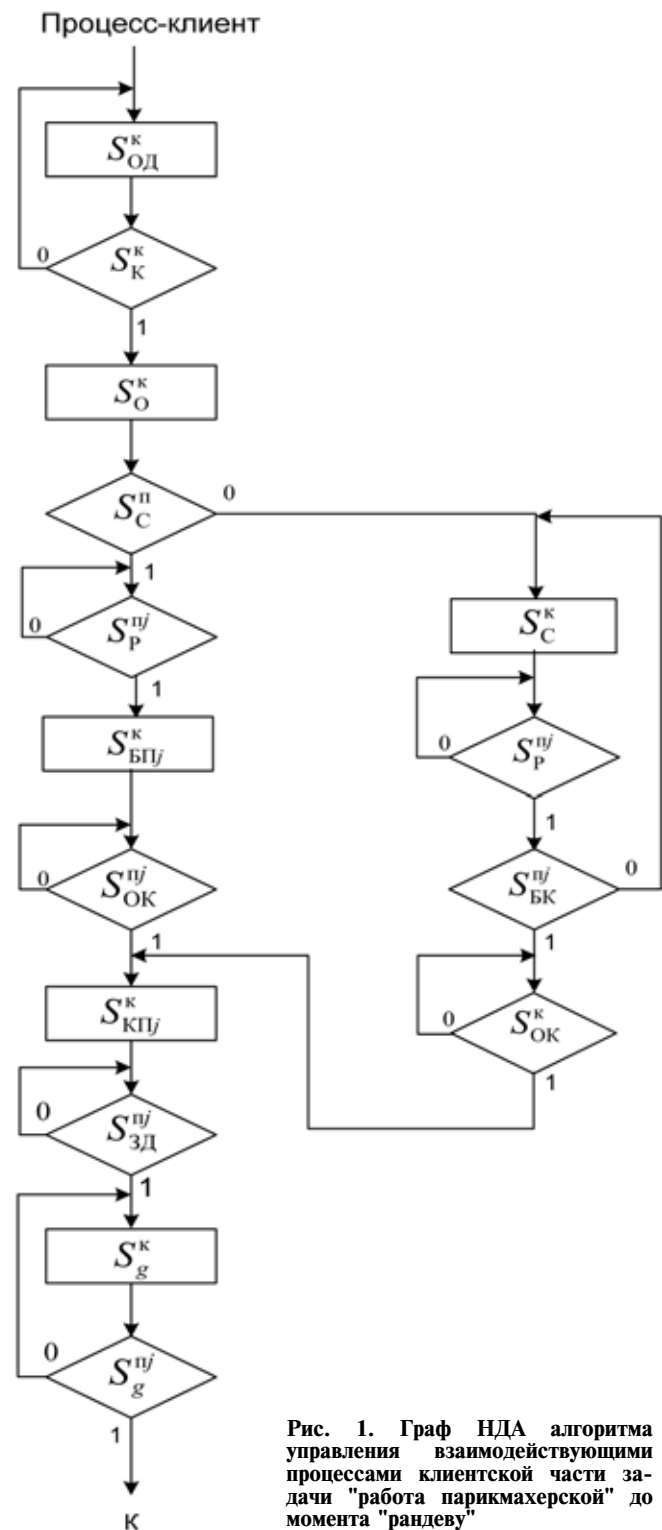
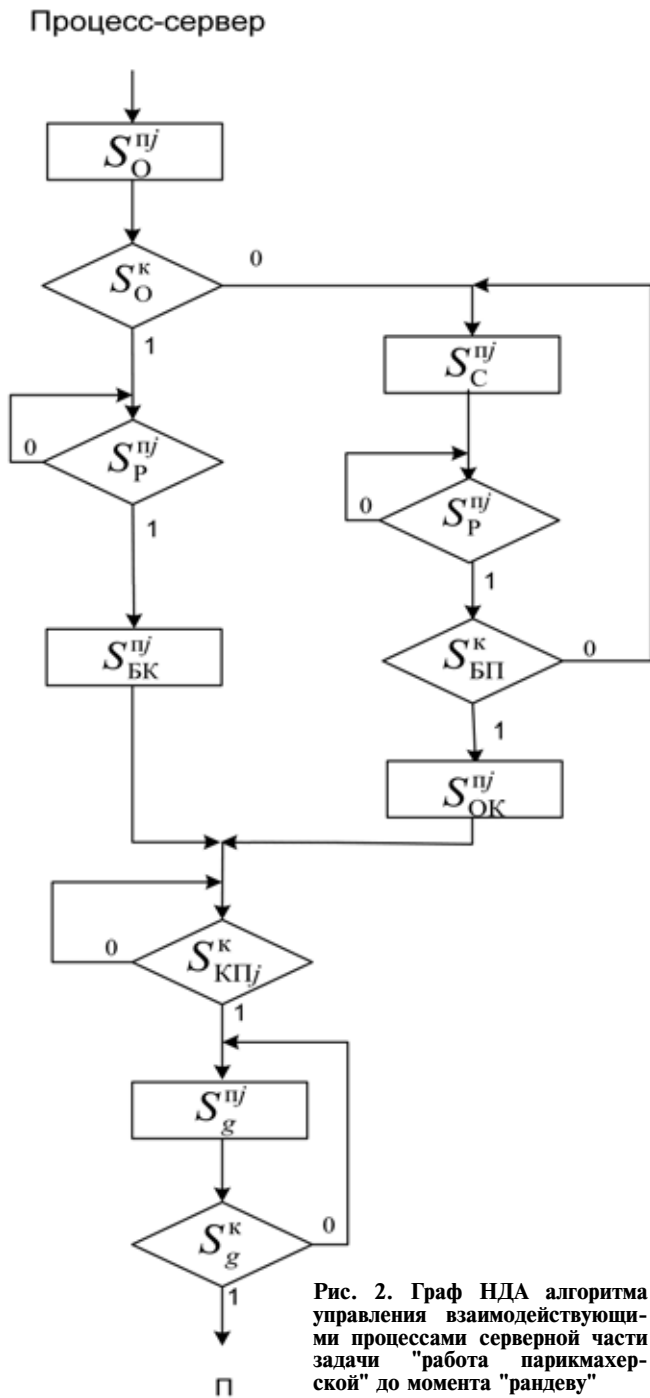


Рис. 1. Граф НДА алгоритма управления взаимодействующими процессами клиентской части задачи "работа парикмахерской" до момента "randevu"

Синхронизация параллельных процессов в данном случае осуществляется на основе использования монитороподобного механизма [5]. Описываемые события связаны с выбором j -го парикмахера для обслуживания очередного i -го клиента с учетом их взаимоисключения и приоритетов. Система уравнений для событий, определяющих



входы для одного из процессов в монитор, будет иметь вид

$$S_M^{nj}(t+1) = S_{ВП}^{nj} S_{ВЗ}^{nj} S_{ПР}^{nj} \vee S_M^{nj} \overline{S_P^{nj}}, j = \overline{1, N}, (6)$$

где S_M^{nj} — сокращенное обозначение события, определяющего вход и нахождение j -го процесса в мониторе (осуществляется выбор j -го парикмахера для обслуживания очередного клиента);

$S_{ВП}^{nj}$ — событие, определяющее прием заявки для

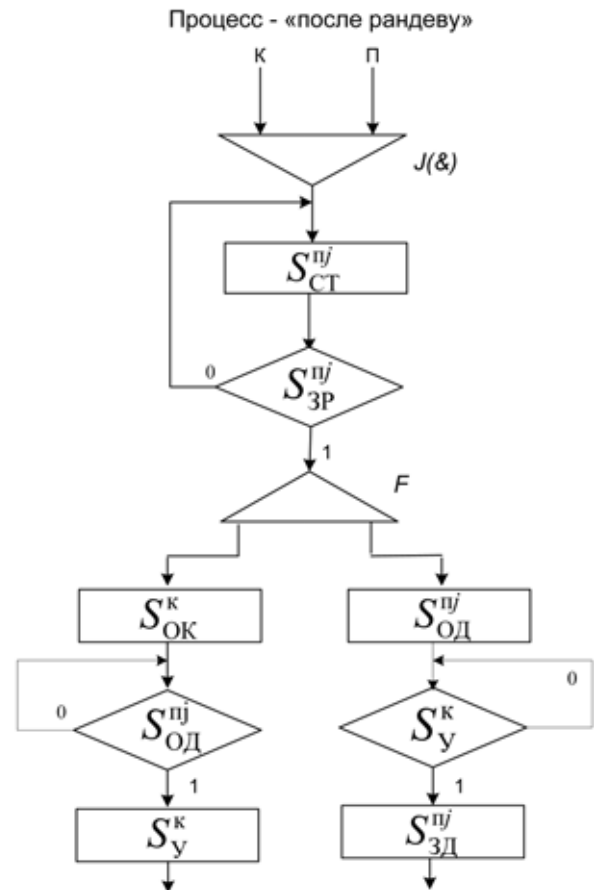


Рис. 3. Граф НДА алгоритма управления взаимодействующими процессами в задаче "работа парикмахерской" после "рандеву"

обращения к разделяемым данным (прием заявки от свободного j -го парикмахера на обслуживание очередного клиента);

$$S_{ВП}^{nj}(t+1) = (S_O^{nj} \vee S_{ВП}^{nj}) \overline{S_M^{nj}}, (7)$$

где S_O^{nj} — событие, определяющее наличие заявки от j -го парикмахера на обслуживание очередного клиента;

S_P^{nj} — событие, свидетельствующее о том, что проведен выбор j -го парикмахера на обслуживание очередного клиента;

$$S_P^{nj}(t+1) = S_M^{nj} S_{ВП}^{nj} \vee S_P^{nj} \overline{S_{БПj}^k}; (8)$$

$S_{ВЗ}^{nj}$ — комбинационное событие, обеспечивающее взаимоисключение процессов выбора свободных парикмахеров для обслуживания очередного клиента на основе несовместимости события S_M^{nj} с другими событиями из их общего числа, равного N :

$$S_{ВЗ}^{nj} = \overline{S_M^{n1}} \overline{S_M^{n2}} \dots \overline{S_M^{nj-1}} \overline{S_M^{nj+1}} \dots \overline{S_M^{nN}}; (9)$$

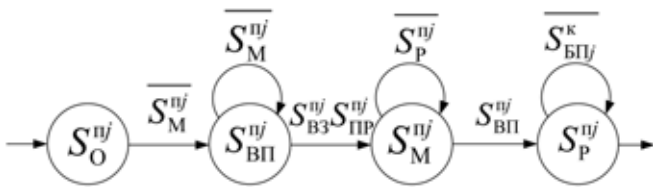


Рис. 4. Граф НДА алгоритма управления взаимодействующими процессами в задаче "выбор парикмахера"

$S_{ПР}^{nj}$ — событие, обеспечивающее приоритетный выбор j -го парикмахера на обслуживание очередного клиента. Алгоритм вычисления приоритетного события для циклической дисциплины обслуживания дан в [7].

Граф НДА, представляющий алгоритм такого взаимодействия, будет иметь вид, показанный на рис. 4.

Заключение

В результате проведенных исследований по формализации алгоритмов многопроцессорных вычислительных систем получены следующие результаты.

- Получена модель, описывающая процедуру синхронизации параллельных процессов с использованием механизма "рандеву" и примити-

ва взаимного исключения типа "монитор" на основе логики недетерминированных автоматов.

- Полученные уравнения для событий, имеющих место при выполнении процедуры синхронизации, могут быть использованы для схемного синтеза блока управления синхронизацией, входящего в состав диспетчера задач многопроцессорной системы.

Список литературы

1. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002. 608 с.
2. Столлингс В. Операционные системы: Пер. с англ. 4-е изд. М: Издат. дом "Вильямс", 2002. 848 с.
3. Вашкевич Н. П. Недетерминированные автоматы в проектировании систем параллельной обработки: Учеб. пособие. Пенза: Изд-во Пензенского гос. ун-та, 2004. 280 с.
4. Эндрюс Г. Р. Основы многопоточного, параллельного и распределенного программирования: Пер. с англ. М.: Издат. дом "Вильямс", 2003. 512 с.
5. Дейтел Г. Введение в операционные системы: Пер. с англ. В 2-х томах, т. 1. М.: Мир, 1987. 359 с.
6. Вашкевич Н. П., Тараканов А. А. Формализация алгоритма управления синхронизацией процессов в задаче о спящем парикмахере. // Межвузовский сборник научных трудов "Вычислительные системы и технологии обработки информации", 2006. С. 6—9.
7. Вашкевич Н. П., Бикташев Р. А., Гурин Е. И. Аппаратная реализация функций синхронизации параллельных процессов при обращении к разделяемому ресурсу на основе ПЛИС // Изв. вузов. Сер. "Технические науки". Поволжский регион. 2007. № 2. С. 3—12.

УДК 004.75; 004.942

Д. А. Сериков, аспирант,

Московский государственный университет имени М. В. Ломоносова

К математическому моделированию процесса диспетчеризации задач в распределенной вычислительной среде

Рассматриваются подходы к математическому моделированию процесса разделения ресурсов между задачами различных пользователей, которые поступают на обслуживание в распределенную вычислительную среду, построенную на основе методологии GRID. Анализируются модели системы диспетчеризации, которые используются в инструментальном комплексе GridWay.

Ключевые слова: Grid, диспетчеризация, моделирование.

Введение

Несмотря на быстрый рост производительности отдельных вычислительных установок [1] решить с их помощью целый ряд сложных научно-технических и практически значимых задач в настоящее время не представляется возможным. Одна из главных причин такого положения дел связана с недостатком необходимых для этого

объемов вычислительных ресурсов¹, которыми даже такие сверхвысокопроизводительные установки не располагают. Вместе с тем высокие темпы развития информационно-вычислительных и

¹ Под ресурсом понимается средство вычислительной машины (или нескольких машин), которым она может пользоваться в процессе своей работы (процессорное время, оперативная и дисковая память и т. п.).

коммуникационных технологий, сетевой инфраструктуры на основе пакетных коммуникаций создают технические предпосылки для консолидации таких ресурсов. Методы построения подобных распределенных (в том числе — географически) информационно-вычислительных комплексов, объединяющих вычислительные ресурсы различных организаций для решения ресурсоемких задач, начали активно развиваться около 10—15 лет назад. В эти годы методология *Grid* применялась для решения многих научных задач, например, связанных с расшифровкой генома человека [2]. В 1997 г. появился первый крупный проект по использованию ресурсов компьютеров обычных пользователей с помощью сети Интернет для решения исследовательских задач, требующих больших вычислительных ресурсов, — проект *distributed.net* [3].

Согласно трем критериям классического определения *Grid*-системы, которые предложил один из идеологов данной методологии Ян Фостер [4], *Grid* — открытая и стандартизованная компьютерная среда, которая обеспечивает децентрализованное разделение ресурсов и высококачественное обслуживание пользователей в рамках виртуальной организации. Под виртуальной организацией понимается группа пользователей, как отдельных лиц, так и структурных подразделений различных форм собственности, совместно использующих общие ресурсы. Открытость и стандартизованность означает тот факт, что система должна строиться на основе стандартных, открытых протоколов и интерфейсов, позволяющих решать такие традиционные задачи как аутентификация, авторизация, обнаружение ресурсов и управление доступом к ним. Кроме того, система должна координировать использование ресурсов при отсутствии централизованного управления ими. Здесь нужно подчеркнуть принципиальное отличие от *Grid*-системы слабосвязанного кластера как группы компьютеров, объединенных высокоскоростными каналами связи и представляющих с точки зрения пользователя единый аппаратный ресурс. В случае такого кластера речь идет о компьютерах, изначально управляемых из единого центра, в случае *Grid* — об одноранговой сети с независимыми узлами. В этом смысле слабосвязанные кластеры не являются *Grid*-системами. Использование ресурсов в *Grid* должно осуществляться таким образом, чтобы обеспечивалось высокое качество обслуживания. Такое обслуживание характеризуется доступностью ресурсов, надежностью работы системы в целом, временем отклика на запрос пользователя, пропускной способностью сетевых каналов и другими подобными им атрибутами.

В силу многопользовательского характера *Grid* необходимым механизмом, обеспечивающим вы-

сокачественное обслуживание, является планирование ресурсов (в рамках процесса диспетчеризации — автоматического распределения ресурсов при обслуживании запросов пользователей). Процесс планирования координирует разделение ресурсов между задачами пользователей. При правильной организации процесс диспетчеризации (и планирования) должен требовать от пользователя минимального участия, а именно, пользователь лишь должен запускать задачу и получать результат. Где и как будет исполняться приложение, должна решить специальная служба, именуемая диспетчером. Диспетчер является одним из основных компонентов любого *Grid*-комплекса, а его функциональность является фактором, определяющим производительность распределенных вычислений. По этой причине разработка математической модели эффективного диспетчера для *Grid*-комплексов является актуальной задачей. В настоящей работе изложены возможные подходы к моделированию процесса диспетчеризации задач, к которым относятся аналитический и дискретно-событийный.

Ключевые понятия

В целях унификации используемых далее терминов и определений введем ключевые понятия.

- *Grid*-среда — открытая, стандартизованная², распределенная вычислительная среда, объединяющая совокупность распределенных ресурсов, которая поддерживает их децентрализованное разделение для решения широкого класса задач с высоким качеством обслуживания в рамках виртуальной организации.
- *Ресурс* — логическая или физическая часть распределенной вычислительной среды *Grid*, которая может быть выделена пользователю. Рассматриваются только вычислительные ресурсы, на которых происходит исполнение задач. Примерами вычислительных ресурсов являются процессорное время, оперативная и дисковая память.
- *Задача* — программная единица для обработки в распределенной вычислительной среде *Grid*. Она включает в себя исполняемый файл, паспорт задачи (файл, содержащий описание задачи), а также файлы входных и выходных данных. Рассматриваются не все возможные задачи, а некоторый подкласс.
- *Виртуальная организация* — владельцы (поставщики) и пользователи (потребители) ресурсов,

² Под стандартизованностью в данной работе понимается тот факт, что среда должна строиться на основе стандартных протоколов и интерфейсов.

действующие на основании некоторых правил предоставления/потребления ресурсов.

- *Диспетчеризация задач* — автоматическая обработка набора задач, включающая в себя *планирование ресурсов*, доставку необходимых входных файлов на ресурс, запуск, управление и мониторинг выполнения задач, доставку выходных файлов.
- *Планирование ресурсов* — распределение задач по доступным ресурсам.

В инфраструктурах *Grid* для диспетчеризации задач могут быть использованы различные подходы. Наиболее распространенный из них — централизованная диспетчеризация задач. При использовании такого подхода все задачи распределяются на ресурсы одним диспетчером. Однако централизация противоречит самой методологии *Grid*, которая основывается на децентрализованном разделении ресурсов. В этой связи диспетчеризация задач в *Grid*-среде должна осуществляться с помощью нескольких независимых диспетчеров. Как следствие, децентрализованный подход к диспетчеризации задач может строиться на базе механизмов подхода централизованного. Централизованная диспетчеризация применяется в системах, описанных в работах [5] и [6]. В данной работе в качестве объекта для исследования рассматривается диспетчер *GridWay* [6], как наиболее распространенный в настоящее время диспетчер для *Grid*-систем. В качестве инфраструктурной основы для исследований функциональных возможностей и эффективности механизмов планирования ресурсов для решения вычислительных задач на *Grid*-среде использовался экспериментальный *Grid*-полигон [7]. Данный полигон был развернут на базе высокопроизводительных вычислительных систем Института механики МГУ (НИИ механики МГУ, Москва) / Института проблем информационной безопасности МГУ (ИПИБ МГУ, Москва), Научно-образовательного Центра компьютерного моделирования и безопасных технологий (НОЦКМиБТ, Москва), а также Института вычислительной математики и математической геофизики Сибирского отделения РАН (ИВМиМГФ СО РАН, Новосибирск), объединенных с помощью сетей передачи данных МГУ—РАН.

1. Аналитическая модель системы диспетчеризации задач, используемой в *GridWay*

Далее описана аналитическая модель системы диспетчеризации задач, которая используется в *GridWay*. Содержательно эта модель отражает количественную зависимость эффективности про-

цесса диспетчеризации задач от параметров отмеченного выше *Grid*-полигона. К их числу относятся период поиска новых узлов в составе *Grid*-полигона, период обновления сведений по ресурсам, период опроса состояния задач и другие, подобные им параметры. Аналитический характер модели означает, что она представляет собой математическую зависимость выходных параметров (эффективность процесса диспетчеризации задач) от входных (параметры *Grid*-полигона). Для формализации эффективности процесса диспетчеризации задач вводятся понятия *вычислительного ресурса* и *функции утилизации вычислительного ресурса*.

Определение 1.1 *Вычислительный ресурс* — логическая или физическая часть распределенной вычислительной среды *Grid*, которая соответствует одной вычислительной машине или кластеру³. Вычислительный ресурс может включать в себя один или несколько процессоров.

Определение 1.2 *Функция утилизации процессора* p_j на временном интервале $(t_1; t_2)$ определяется как отношение числа тактов процессора, потраченных на пользовательский режим⁴ на временном интервале $(t_1; t_2)$, к общему числу тактов процессора на этом интервале:

$$f_j(t_1, t_2) = \frac{U_j(t_2) - U_j(t_1)}{T_j(t_2) - T_j(t_1)}.$$

Содержательно данное понятие означает тот факт, что чем меньше разность $1 - f_j(t_1, t_2)$, тем эффективнее используется процессор p_j . Без ограничения общности можно считать, что моделирование проводится на процессорах с постоянной тактовой частотой $T(t) = kt$, где k — тактовая частота процессора. Для таких процессоров функция утилизации приобретает вид

$$f_j(t_1, t_2) = \frac{U_j(t_2) - U_j(t_1)}{k_j(t_2 - t_1)}.$$

Определение 1.3 *Функция утилизации вычислительного ресурса* h_i на временном интервале $(t_1; t_2)$ представляет собой среднее арифметическое функций утилизации процессоров p_j на соответствующую

³ В контексте данной работы термином *кластер* (слабосвязанный кластер) будем именовать группу компьютеров, объединенных высокоскоростными каналами связи, представляющую с точки зрения пользователя единый аппаратный ресурс.

⁴ Режим *user mode* — непривилегированный режим работы процессора, предназначенный для исполнения прикладных программ. В отличие от привилегированного режима (*system mode*) доступ к системным ресурсам программ предоставляется только посредством системных вызовов.

щем временном интервале для всех процессоров p_j , входящих в ресурс h_i :

$$F_i(t_1, t_2) = \frac{1}{n_i} \sum_{p_j \in h_i} f_j(t_1, t_2) = \\ = \frac{1}{n_i(t_2 - t_1)} \sum_{p_j \in h_i} \frac{U_j(t_2) - U_j(t_1)}{k_j},$$

где n_i — число процессоров узла h_i .

В операционной системе GNU/Linux величину $U(t)$ в любой момент времени можно узнать с помощью файла `/proc/stat`. Однако массовое вычисление этой функции (например, на многопроцессорных кластерах) потребовало бы разработки собственной системы мониторинга. Существующие же системы мониторинга могут вычислять несколько иную величину.

Определение 1.4 Загрузка процессора p_j в момент времени t есть значение функции утилизации процессора p_j на временном интервале $(t - \Delta t, t)$:

$$c_j(t) = c_j(t, \Delta t) = \frac{U_j(t) - U_j(t - \Delta t)}{k_j \Delta t},$$

где Δt — фиксированный отрезок времени.

Предположим, что $t_2 - t_1 = n\Delta t$, где n — целое, тогда $U_j(t_2) - U_j(t_1) = k_j \Delta t (c_j(t_2) + c_j(t_2 - \Delta t) + \dots + c_j(t_2 - (n-1)\Delta t))$.

Таким образом функция утилизации для вычислительного ресурса h_i на временном интервале $(t_1; t_2)$ рассчитывается по формуле

$$F_i(t_1, t_2) = \frac{\Delta t}{n_i(t_2 - t_1)} \sum_{p_j \in h_i} \sum_{l=0}^{n-1} c_j(t_2 - l\Delta t).$$

В целях получения количественной характеристики эффективности системы диспетчеризации задач, используемой в *GridWay*, было проведено несколько последовательных запусков задач на *Grid*-полигоне [7].

Для измерения загрузки процессора в зависимости от компьютерной системы можно использовать пакеты `collectd` [8] и `Ganglia` [9]. Отличительной особенностью этих пакетов является тот факт, что собранная ими информация помещается в `RRD`-файлы (`Round Robin Database` [10]). С помощью набора утилит для работы с `RRD` (`RRDtool`) информация о системной активности может быть извлечена и проанализирована. Сбор информации с удаленных хостов можно осуществить с помощью `SSH`.

Тестирование осуществлялось следующим образом. На выполнение посылались большой пул задач (1000), а функция утилизации для каждого ресурса вычислялась периодически через каждые 15 мин.

Вычисление функции утилизации

SI	DI	MI	PI	DC	tp(2), %	gf(2), %	imec(16), %	sscc(81), %
30	900	300	180	15	62,58	75,76	66,31	51,36
15	900	300	180	15	65,82	81,00	75,99	53,38
60	900	300	180	15	53,98	69,88	63,75	51,31
30	450	300	180	15	66,70	80,36	61,18	45,51
30	1800	300	180	15	62,50	70,80	74,22	53,46
30	900	150	180	15	65,66	72,93	62,87	53,17
30	900	600	180	15	53,75	81,90	68,07	49,98
30	900	300	90	15	63,61	82,75	67,00	61,34
30	900	300	360	15	56,91	75,42	60,10	42,15
30	900	300	180	5	41,53	48,39	45,23	35,49
30	900	300	180	30	69,33	79,34	68,36	53,24

Выявление основных параметров *Grid*-полигона, которые в наибольшей степени влияют на эффективность диспетчеризации задач. Диспетчер *GridWay* имеет несколько параметров (все они задаются в конфигурационных файлах `gwd.conf` и `sched.conf`), комбинация которых, принятая для исполнения, влияет на эффективность процесса диспетчеризации. К основным из них относятся: `SCHEDULING_INTERVAL` (*SI*) — период между двумя итерациями планирования; `DISCOVERY_INTERVAL` (*DI*) — период поиска новых хостов в *Grid*-полигоне; `MONITORING_INTERVAL` (*MI*) — период обновления сведений по каждому ресурсу; `POLL_INTERVAL` (*PI*) — период опроса о состоянии задач; `DISPATCH_CHUNK` (*DC*) — максимальное число задач, которые будут обработаны планировщиком за один шаг планирования. В таблице приведены результаты вычисления значений функции утилизации четырех ресурсов *Grid*-полигона, полученные при указанных параметрах.

В качестве вычислительного приложения использовались: доказательство ненадежности криптоалгоритма `RC5` путем взлома зашифрованного им сообщения; поиск новых, более оптимальных "линеек Голомба" [10]. Обе задачи представлены в материалах проекта распределенных вычислений `distributed.net` [3].

Результаты тестирования показали, что из перечисленных пяти параметров в наибольшей степени на эффективность процесса диспетчеризации задач влияют *SI*, *PI* и *DC*. Функция утилизации при этом растет с уменьшением периода между двумя итерациями планирования (*SI*), а также с сокращением периода опроса ресурсов о состоянии выполняемых на них задач (*PI*). Низкие значения параметра *DC* значительно снижают эффективность процесса диспетчеризации, что неудивительно — за одну итерацию планирования задач обрабатывается меньше, чем доступно ресурсов. Как результат — ресурсы начинают "простаивать". К сожалению, тестирование не позволило определить зависимости функции утилизации от периода поиска новых хостов в составе

Grid-полигона (DI) и периода обновления сведений по каждому ресурсу (MI). При последовательном уменьшении (увеличении) этих параметров значение функции утилизации могло, как уменьшаться, так и увеличиваться.

2. Имитационное моделирование

Представленная ранее аналитическая модель основана на анализе эффективности процесса диспетчеризации задач в различных условиях. К ним относятся разные значения входных параметров, в том числе характеризующих состояние полигона и системы диспетчеризации. Такая модель позволяет получить адекватный результат, поскольку опирается на практические данные. Вместе с тем подобный подход требует очень больших вычислительных затрат на его реализацию. По этой причине целесообразно исследовать систему с помощью методов *имитационного моделирования* [11]. Такой способ моделирования позволяет проводить многократные испытания модели с нужными входными данными, чтобы определить их влияние на выходные критерии оценки работы системы.

Один из видов имитационного моделирования — *дискретно-событийное моделирование*. Оно используется для построения модели, отражающей развитие системы во времени, когда состояние системы меняется мгновенно, в конкретные моменты времени. Смену состояния системы называют *событием*. Одна из наиболее распространенных моделей дискретно-событийного моделирования — *система массового обслуживания* — объект, деятельность которого связана с многократной реализацией исполнения некоторых однотипных задач и операций. Система массового обслуживания состоит из обслуживаемой и обслуживающей подсистем. Обслуживаемая подсистема включает совокупность источников требований⁵ и входящего потока требований. Обслуживающая система состоит из накопителя и механизма обслуживания.

Учитывая специфику исследуемой системы диспетчеризации задач, которая, по сути, является системой, обслуживающей входящий поток задач, можно предположить, что эффективным для нее будет применение модификации стандартной дискретно-событийной модели системы массового обслуживания.

2.1. Моделирование систем массового обслуживания

Система массового обслуживания состоит из одного или нескольких устройств обслуживания,

⁵ В теории массового обслуживания русскоязычный термин *требование* является устоявшимся аналогом английского термина *customers*.

которые предоставляют какие-либо услуги входящему потоку требований. Требования, как правило, поступающие, когда все устройства заняты, образуют одну или несколько очередей к устройствам обслуживания.

Система массового обслуживания включает в себя три компонента: процесс поступления; механизм обслуживания; дисциплину обслуживания.

Процесс поступления состоит из описания механизма появления требований в системе массового обслуживания.

Механизм обслуживания в системе массового обслуживания определяется следующими факторами: числом устройств обслуживания (обозначаемым как N); наличием для каждого устройства своей очереди или существованием одной очереди для всех устройств; распределением вероятностей времени обслуживания требований.

Дисциплина обслуживания определяется правилом, которое устройство обслуживания использует для выбора из очереди следующего требования (если таковые есть) по завершении обслуживания требования текущего. Обычно используются следующие дисциплины очереди:

- *требования FIFO (First-In, First-Out)*, они обслуживаются по принципу "первым поступило — первым обработано";
- *требования LIFO (Last-In, First-Out)*, они обслуживаются по принципу "последним поступило — первым обработано";
- *требования с приоритетом*, которые обслуживаются в порядке их значимости или в соответствии с другими требованиями к обслуживанию.

Рассмотрим систему массового обслуживания, которая имеет перечисленные ниже характеристики.

1. N устройств массового обслуживания и одна очередь FIFO для всех этих устройств.

2. A_1, A_2, \dots — независимые и одинаково распределенные величины, обозначающие время между поступлениями требований.

3. S_1, S_2, \dots — независимые и одинаково распределенные величины, обозначающие время обслуживания поступивших требований.

4. Величины A_i и S_i не зависят друг от друга.

Такая система массового обслуживания обозначается как $GI/G/N$, при этом GI (*General Independent* — произвольное независимое) относится к распределению величин A_i , а G (*General* — произвольное) — к распределению величин S_i .

2.2. Дискретно-событийная модель системы диспетчеризации задач, используемой в GridWay

Рассмотрим систему массового обслуживания $GI/G/N$ с N устройствами обслуживания. В этой терминологии система массового обслуживания

соответствует диспетчеру *GridWay*, а каждое из устройств обслуживания — вычислительному ресурсу⁶. В этой системе интервалы времени между поступлениями требований (задач) A_1, A_2, \dots являются независимыми, одинаково распределенными случайными величинами. Когда задача поступает, и при этом хотя бы один из ресурсов свободен, обслуживание задачи начинается немедленно. Время обслуживания S_1, S_2, \dots следующих задач представлено также независимыми одинаково распределенными случайными величинами. Если при поступлении задачи все ресурсы заняты, то задача ставится в очередь. При освобождении какого-либо ресурса (то есть по завершении обслуживания одной из задач) диспетчер выбирает задачу из очереди (если такая имеется) по принципу *FIFO* (*First-In, First-Out* — "первый пришел — первый вышел").

Моделирование начинается с состояния, когда в системе нет задач и все ресурсы свободны. С момента времени, равного нулю (начального), начинаем ждать поступления первой задачи. С большой долей вероятности это произойдет по истечении некоторого интервала времени A_1 , а не в начальный момент времени. Моделирование завершается по прошествии фиксированного периода времени T .

Далее рассмотрим видоизмененную систему массового обслуживания $GI/G/N$, которая имеет следующие свойства.

1. Если хотя бы один из ресурсов свободен, то выборка задач из очереди и их обслуживание начинается не немедленно, а в моменты времени, кратные параметру SI , где $SI \geq 0$ — параметр системы.

Применительно к диспетчеру *GridWay* этот факт означает то, что планирование ресурсов для задачи происходит не в момент ее поступления, а периодически, с интервалом в SI секунд, что соответствует параметру `SCHEDULING_INTERVAL`.

2. Величины N_0, N_1, N_2, \dots — независимые и одинаково распределенные, представляющие общее число ресурсов системы в моменты времени $0, DI, 2DI, \dots$, где $DI \geq 0$ — параметр системы.

Параметр DI соответствует периоду времени, с которым диспетчер *GridWay* проверяет *Grid*-полигон на наличие новых ресурсов (параметр `DISCOVERY_INTERVAL` диспетчера).

3. Величины M_0, M_1, M_2, \dots — независимые и одинаково распределенные, представляющие число ресурсов системы, занятых локальными очередями соответственно в моменты времени $0, MI, 2MI, \dots$, где $MI \geq 0$ — параметр системы.

Параметр MI соответствует периоду обновления сведений по каждому ресурсу (параметр `MONITORING_INTERVAL` диспетчера).

4. Если обслуживание задачи i началось в момент времени kSI (п. 1), то соответствующий ресурс освободится в момент времени $PI \left\lceil \frac{kSI + S_i}{PI} \right\rceil$, где $PI \geq 0$ — параметр системы.

Диспетчер *GridWay* опрашивает ресурсы о состоянии задач не постоянно, а периодически, с интервалом PI (соответствующий параметр `POLL_INTERVAL`). По этой причине после выполнения задачи ресурс сможет освободиться только в момент времени, кратный PI .

5. За один шаг планирования, соответствующий выборке задач из очереди, из нее извлекается не более DC задач, где $DC > 0$ — параметр системы.

Параметр DC соответствует максимальному числу задач, которые будут обработаны модулем планирования *GridWay* за одну итерацию (параметр `DISPATCH_CHUNK`) диспетчера).

Определение 2.1 Система с перечисленными выше свойствами 1—5 называется системой массового обслуживания $GI/G/D/M$ (SI, DI, MI, PI, DC). При этом GI относится к распределению величин A_i ; G — к распределению величин S_i ; D и M — к распределению величин N_i и M_i , соответственно.

Заметим, что система массового обслуживания $GI/G/D/M$ (SI, DI, MI, PI, DC) является дискретно-событийной моделью системы диспетчеризации задач, используемой в *GridWay*. Кроме того, при $N_0 = N_1 = N_2 = \dots = N$, $M_0 = M_1 = M_2 = \dots = 0$ система массового обслуживания $GI/G/N$ является предельным случаем системы $GI/G/D/M$ (SI, DI, MI, PI, DC) при $SI, PI \rightarrow 0, DC \rightarrow \infty$.

2.3. Критерии оценки работы системы диспетчеризации задач, используемой в *GridWay*

Существует достаточно большое число критериев оценки работы систем массового обслуживания. Рассмотрим те показатели эффективности работы системы диспетчеризации, которые обычно используются при математическом исследовании систем массового обслуживания. Примем следующие обозначения: D_i — время задержки в очереди задачи i ; $Q(t)$ — число задач в очереди в момент времени t ; $B(t)$ — число занятых ресурсов.

Тогда показатель

$$d = \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n D_i$$

с вероятностью⁷ 1 (если он существует) называется *установившейся средней задержкой*. Соответственно показатель

⁷ Здесь выражение "с вероятностью 1" дано для соблюдения математической корректности и не имеет практического значения, поэтому далее будет опускаться.

⁶ Определение вычислительного ресурса дано в разд. 1.

$$Q = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T Q(t) dt$$

(если он существует) называется *установившимся средним по времени числом задач в очереди*.

Показатель

$$u = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T B(t) dt$$

(если он существует) называется *установившимся коэффициентом использования ресурсов* (часть времени, когда ресурсы находятся в состоянии "занято").

2.4. Компьютерное моделирование дискретно-событийной модели системы диспетчеризации задач, используемой в GridWay

Далее определим компоненты, необходимые для компьютерного моделирования системы массового обслуживания $GI/G/D/M$ (SI, DI, MI, PI, DC) на языке С.

Временные интервалы между поступлениями задач и время обслуживания будут моделироваться величинами, подчиняющимися экспоненциальным распределениям, как независимые случайные величины. Экспоненциальное распределение со средним значением $\lambda > 0$ является непрерывным с плотностью распределения вероятностей [10]

$$f(x) = \frac{1}{\lambda} e^{-x/\lambda}.$$

Экспоненциальное распределение выбрано по той причине, что на компьютере несложно сгенерировать экспоненциально распределенные случайные величины. В действительности допущение экспоненциально распределенного времени между поступлениями запросов на обслуживание, как правило, хорошо соответствует реалиям. Принятие экспоненциально распределенного времени обслуживания задач при этом — менее правдоподобно [11].

Определение 2.2 Систему массового обслуживания $GI/G/D/M$ (SI, DI, MI, PI, DC) с экспоненциально распределенными временными интервалами между поступлением задач, временем их обслуживания, общим числом ресурсов и числом ресурсов, занятых локальными очередями, будем называть системой $M/M/M/M$ (SI, DI, MI, PI, DC) (M обозначает экспоненциальное распределение).

Для генерирования случайных чисел будем использовать метод Марсе и Робертса [12].

Программа состоит из нескольких подпрограмм. Кроме основной программы, программа моделирования включает в себя подпрограммы

для инициализации, синхронизации, обработки событий, генерирования отчетов и экспоненциально распределенных случайных величин. Самые важные действия выполняются в подпрограммах событий и имеют следующую нумерацию: поступление задачи в систему (1); уход задачи из системы по окончании обслуживания (2); планирование задач (3); поиск новых ресурсов (4); обновление информации о ресурсах (5); генерация отчета (6). Моделирование завершается тогда, когда число задач, задержанных в очереди, достигает 1000.

Код программы можно найти на сайте [13].

2.5. Анализ выходных данных моделирования

Далее кратко прокомментируем некоторые результаты компьютерного моделирования системы массового обслуживания $M/M/M/M$ (SI, DI, MI, PI, DC), полученные в ходе экспериментов. Моделирование проводилось при следующих параметрах конфигурации системы: среднее время ме-

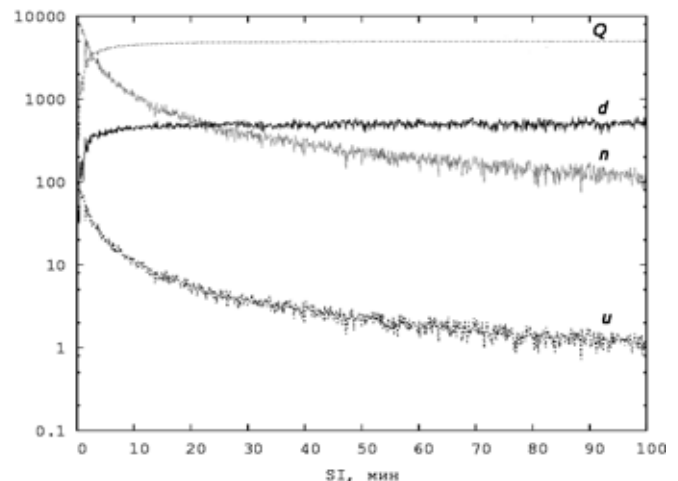


Рис. 1. Зависимость показателей эффективности d (мин), Q (шт.), u (узлы) и n (шт.) от SI

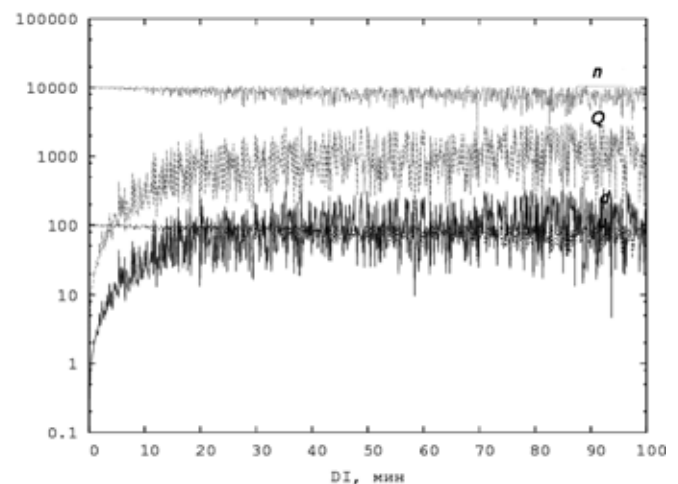


Рис. 2. Зависимость показателей эффективности d (мин), Q (шт.), u (узлы) и n (шт.) от DI

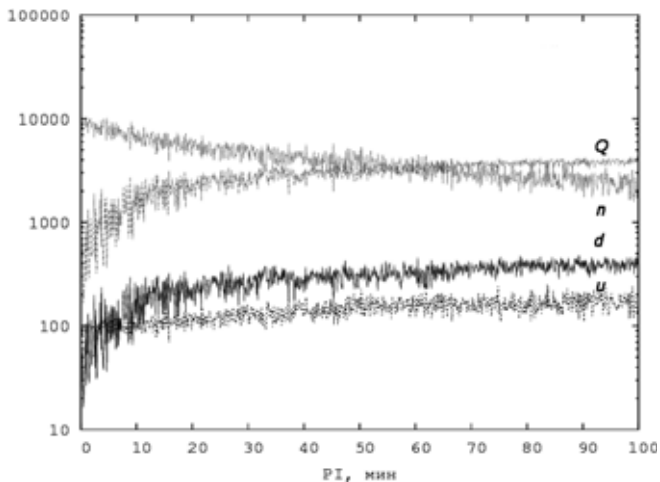


Рис. 3. Зависимость показателей эффективности d (мин), Q (шт.), u (узлы) и n (шт.) от PI

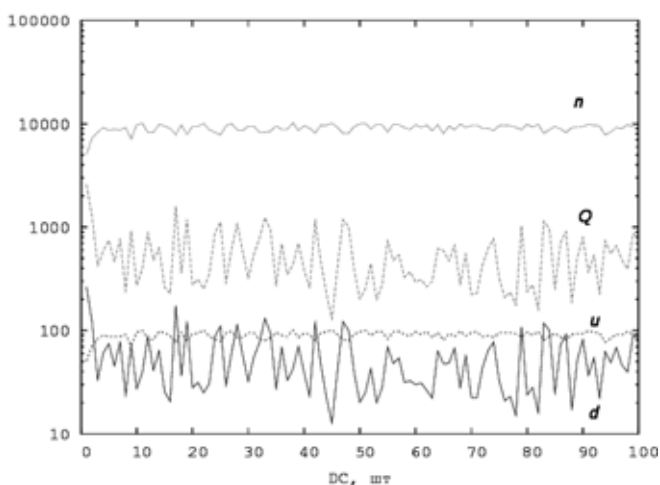


Рис. 4. Зависимость показателей эффективности d (мин), Q (шт.), u (узлы) и n (шт.) от DC

жду поступлениями — 0,1 мин; среднее время обслуживания — 10 мин.; среднее число ресурсов — 100 шт.; среднее число локально занятых ресурсов — 20 шт.; $SI = 0,1$ мин; $DI = 15$ мин; $MI = 5$ мин; $PI = 0,1$ мин; $DC = 15$ шт. В процессе моделирования менялся один из параметров при фиксированных остальных.

При увеличении параметра SI коэффициент использования ресурсов и общее число обслуженных задач падает (рис. 1). Отсюда следует, что оптимальным значением параметра SI является наименьшее допустимое его значение.

Как следует из рис. 2, параметр DI практически не оказывают никакого влияния на эффективность модели. Аналогичный результат был получен также и для параметра MI .

Результаты экспериментов, представленные на рис. 3, свидетельствуют о том, что увеличение параметра PI ведет к падению коэффициента использования ресурсов и общего числа задач, по-

лучивших обслуживание. В этой связи оптимальным значением параметра PI является его наименьшее допустимое значение.

Как следует из рис. 4, малые значения параметра DC уменьшают эффективность модели. Как следствие, значение этого параметра должно быть достаточно большим.

Заключение

Работа посвящена подходам к математическому моделированию процесса диспетчеризации задач в *Grid*-среде. Рассмотрены два из возможных методов моделирования — аналитический и дискретно-событийный (имитационное моделирование). Приведены результаты компьютерного моделирования каждого из изложенных подходов.

В отличие от аналитической модели дискретно-событийный подход не требует очень больших вычислительных затрат на свою реализацию. Для сравнения, около 6000 запусков модели (в каждом запуске имитируется обслуживание 1000 задач) выполнилось всего за 2 мин (CPU Pentium 4 3.00GHz). Сбор входных данных для аналитической модели занял несколько суток. В то же время аналитический подход больше приближен к реальности, чем дискретно-событийный, поскольку опирается на практические данные. В этой связи для моделирования процесса диспетчеризации задач целесообразно использовать оба подхода: дискретно-событийный как теоретическую модель, аналитический как практическую проверку этой модели.

В заключение автор выражает благодарность своему научному руководителю, д-ру физ.-мат. наук, проф. В. А. Васенину за постоянное внимание к работе и помощь при подготовке настоящей публикации.

Список литературы

1. **TOP500** Supercomputing Sites. URL: <http://www.top500.org>
2. **Human Genome Project**. URL: http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml
3. **distributed.net**: Node Zero. URL: <http://distributed.net>
4. **Foster I.** What is the Grid? A Three Point Checklist. // Grid Today. 2002. 1 (6).
5. **EGEE JRA1 Workload Management — WMS Architecture overview**. URL: <http://egee-jral-wm.mi.infn.it/egee-jra1-wm/wms.shtml>
6. **GridWay Metascheduler: Metascheduling Technologies for the Grid**. URL: <http://gridway.org>
7. **GRID МГУ**. URL: <http://grid.pp.ru>
8. **Collectd** — The system statistics collection daemon. URL: <http://collectd.org>
9. **Ganglia Monitoring System**. URL: <http://ganglia.sourceforge.net>
10. **Википедия**. URL: <http://ru.wikipedia.org>
11. **Law A., Kelton W.** Simulation Modeling and Analysis, 3rd. New York: McGraw-Hill, 2000.
12. **Marse K., Roberts S. D.** Implementing a portable FORTRAN uniform (0, 1) generator // Simulation. 1983. N 41. P. 135—139.
13. **Severl-resource** queuing system, fixed run length. URL: http://grid.pp.ru/git/?p=serd/chapl_c.git;a=summary

О. Н. Коваленко, канд. техн. наук,
ст. преподаватель,
Омский государственный университет
путей сообщения,
e-mail: onkovalenko@mail.ru

Влияние метода обработки очереди на качество обслуживания трафика мультисервисной сети

Предлагаются математические модели наиболее распространенных способов распределения пропускной способности канала мультисервисной сети между конкурирующими потоками — PQ (приоритетные очереди) и CBWFQ (взвешенная справедливая буферизация, основанная на классах) для системы массового обслуживания M/M/N:n.

Ключевые слова: обслуживание, канал, качество, очередь, приоритет.

В последнее время становится актуальным вопрос создания единой мультисервисной сети связи железнодорожного транспорта на базе сетей с коммутацией пакетов в целях организации единой информационной структуры и интеллектуальной среды для технологических и бизнес-процессов отрасли [1]. При этом любой узел пакетной сети, обеспечивающий высокое качество обслуживания информации различных типов, нуждается в механизме идентификации и классификации трафика по приоритетам, основывающемся на значении QoS (качества обслуживания) [2].

Решить проблему обеспечения качества обслуживания потоков разных типов в условиях ограниченных канальных ресурсов можно, используя различные методы обработки потоков в узлах мультисервисной сети, а также путем классификации поступающих потоков и гибкого распределения ресурсов между очередями.

В работе для анализа выбраны следующие распространенные способы обработки поступающих потоков и разделения пропускной способности канала между конкурирующими потоками: PQ (приоритетные очереди); CBWFQ (взвешенная справедливая буферизация, основанная на классах).

Проведем сравнительный анализ основных параметров QoS для перечисленных методов обслуживания в многоканальной системе массового обслуживания (СМО) с пуассоновским распределением входящего потока, экспоненциальным временем обслуживания при ограниченном буфе-

ре. По системе Кендела эта СМО обозначается $M/M/N:n$.

К основным параметрам QoS , характеризующим работу СМО $M/M/N:n$, относятся время ожидания в очереди, длина очереди и вероятность потерь при переполнении очереди.

При использовании дисциплины обслуживания с приоритетами PQ предположим, что классы сообщений, обозначаемые индексом $p = 1, 2, 3, \dots, r$, пронумерованы в порядке уменьшения приоритета. Для данной дисциплины обслуживания в работе [3] получены аналитические выражения для расчета времени ожидания в очереди при использовании СМО $M/M/1$, но эти результаты не применимы для СМО $M/M/N:n$.

Рассмотрим сообщение с приоритетом p , которое прибывает в систему в некоторый момент времени и через интервал $M(T_{ож})_{pPQ}$ поступает на обслуживание, т. е. начинается его передача по каналу.

Получим выражение для $M(T_{ож})_{pPQ}$ среднего времени ожидания сообщения с приоритетом p для многоканальной СМО с ожиданием и ограниченным буфером. Интервал $M(T_{ож})_{pPQ}$ складывается из следующих компонентов [4]: среднее время $M(T_k)$, необходимое для обслуживания m_k сообщений с приоритетами $k = 1, 2, \dots, p$, уже ожидающих обслуживания в очереди к моменту поступления рассматриваемого сообщения, и среднее время $M(T'_k)$, необходимое для обслуживания сообщений с более высоким приоритетом, которые могут поступить за интервал ожидания и будут обслужены раньше данного сообщения. Суммируя средние значения всех этих случайных величин, получим

$$M(T_{ож})_{pPQ} = \sum_{k=1}^p M(T_k) + \sum_{k=1}^{p-1} M(T'_k), \quad (1)$$

$$\text{где } M(T_k) = \frac{M(m_k)}{\mu_k}; \quad M(T'_k) = \frac{M(m_p)}{\mu_p} \rho_k; \quad (2)$$

$M(m_i)$ — средняя длина очереди i -го приоритета; μ_i — интенсивность обслуживания пакетов i -го приоритета; ρ_i — коэффициент загрузки канала пакетами i -го приоритета.

Подставляя в формулу (1) выражения (2) и упрощая полученную зависимость, приведем окончательное выражение для расчета среднего значения времени ожидания пакета при использовании дисциплины обслуживания PQ:

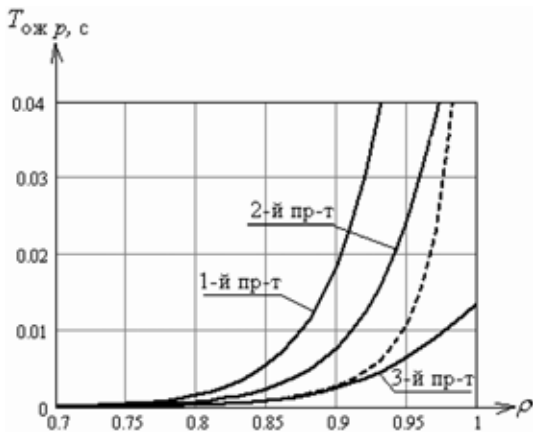


Рис. 1. Зависимость длительности ожидания от загрузки канала (методы обслуживания FIFO и PQ)

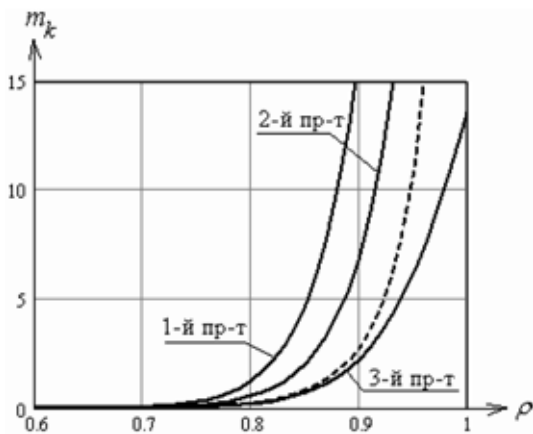


Рис. 2. Зависимость длины очереди от загрузки канала (методы обслуживания FIFO и PQ)

$$M(T_{ож})_{PQ} = \sum_{k=1}^{p-1} \frac{M(m_k)}{\mu_k} + \frac{M(m_p)}{\mu_p} \left(1 + \sum_{k=1}^{p-1} \rho_k\right). \quad (3)$$

Длина очереди в случае рассматриваемой СМО $M/M/N:n$ с ограниченной очередью согласно [5] с учетом приоритетов рассчитывается по формуле

$$M(m_k) = \frac{(N \cdot \rho_k)^N}{N!} \left(\sum_{i=0}^N \frac{(N \cdot \rho_k)^i}{i!} + \frac{(N \cdot \rho_k)^N}{N!} \sum_{S=1}^{n_k} \rho_k^S \right)^{-1} \sum_{S=1}^{n_k} S \rho_k^S, \quad (4)$$

где n_k — емкость буфера k -го приоритета; вероятность переполнения очереди — по формуле

$$p_{пот k} = \frac{(N \cdot \rho_k)^N}{N!} \times \left(\sum_{i=0}^N \frac{(N \cdot \rho_k)^i}{i!} + \frac{(N \cdot \rho_k)^N}{N!} \sum_{S=1}^{n_k} \rho_k^S \right)^{-1} \rho_k^{n_k}. \quad (5)$$

Результаты расчетов по формулам (1)—(5) приведены на рис. 1, 2.

На рис. 1 и 2 штриховой линией для сравнения показана зависимость основных величин от загрузки канала для случая обслуживания FIFO (обслуживание в порядке поступления пакетов без учета приоритетов), сплошной линией — зависимости соответствующих величин при дисциплине обслуживания PQ.

Как видно из расчетов, при использовании дисциплины обслуживания с приоритетами значительно улучшаются показатели QoS для трафика реального времени (приоритет 1). Время, затрачиваемое на обслуживание очереди пакетов 2-го и 3-го приоритетов при использовании дисциплины PQ, увеличилось.

Данный алгоритм имеет такие положительные стороны, как простота реализации и возможность малой задержки для пакетов высокоприоритетного потока, а также отрицательные стороны, одна из которых влияние потоков друг на друга, так как если высокоприоритетная нагрузка будет постоянно поступать в очередь с высокой интенсивностью, то нагрузка остальных очередей будет бло-

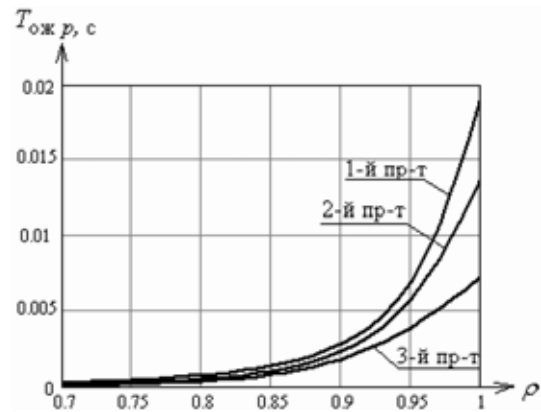


Рис. 3. Зависимость длительности ожидания от загрузки канала (метод обслуживания CBWFQ)

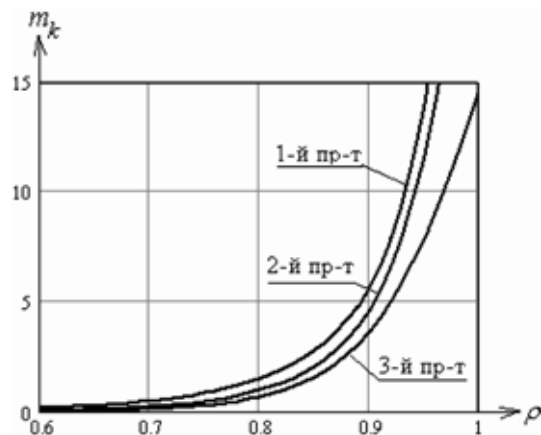


Рис. 4. Зависимость длины очереди от загрузки канала (метод обслуживания CBWFQ)

кирована. Следовательно, применение приоритетной дисциплины обслуживания оправдано, если заранее известно, что высокоприоритетный трафик имеет значительно меньшую интенсивность по сравнению с низкоприоритетным.

Данный отрицательный эффект можно нейтрализовать с помощью выделения каждому типу трафика фиксированной гарантированной полосы пропускания (в примере с помощью *CBWFQ*). В случае использования *CBWFQ* каждый пакет помещается в очередь соответствующего класса (приоритета). При этом для обслуживания каждой очереди выделяется пропускная способность r_i или определенное количество единиц канального ресурса N_k , что позволяет обеспечить минимальные гарантии по качеству обслуживания. В пределах одной очереди пакеты обслуживаются по принципу *FIFO*. Среднее число заявок в очереди определяется из выражения

$$M(m_k)_{CBWFQ} = \frac{(N_k \cdot \rho_k)^{N_k}}{N_k!} \times \left(\sum_{i=0}^{N_k} \frac{(N_k \cdot \rho_k)^i}{i!} + \frac{(N_k \cdot \rho_k)^{N_k}}{N_k!} \sum_{S=1}^{n_k} \rho_k^S \right)^{-1} \sum_{S=1}^{n_k} \rho_k^S, \quad (6)$$

а значение задержки в очереди — по формуле

$$M(T_{ож})_k_{CBWFQ} = \frac{M(m_k)_{CBWFQ}}{\mu_k}. \quad (7)$$

Результаты расчетов по формулам (6), (7) приведены на рис. 3, 4.

Выводы. В данной работе показано, что если нагрузка канала не превышает 0,4, то влияние конкурирующих потоков минимально и можно использовать простейший алгоритм обработки очередей — *FIFO*. При увеличении загрузки канала, но невысокой доле трафика реального времени (около 10—15 % от всего трафика) следует применить алгоритм *PQ*.

В случае, если коэффициент загрузки канала превышает значение 0,4 и трафик реального времени составляет более 20—30 % от общего объема трафика, лучше выбрать методы обслуживания очередей *CBWFQ* или *WFQ*.

Список литературы

1. Голышко А. В., Степанов С. Н., Харкевич А. Д. Новые сети и новые задачи расчета телетрафика // Труды Российского научно-технического общества радиотехники, электроники и связи имени А. С. Попова. Серия: научная сессия, посвященная Дню радио. 2007. Вып. LXII. С. 25—29.
2. Кучерявый Е. А. Управление трафиком и качество обслуживания в сети Интернет. СПб: Наука и техника. 2004. 336 с.
3. Крылов В. В., Самохвалова С. С. Теория телетрафика и ее приложения. СПб: БВХ — Петербург, 2005. 288 с.
4. Коваленко О. Н. Влияние системы поллинга на качество обслуживания трафика // Труды XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых "Современные техника и технологии", 26—30 марта 2007 г. В 3-х томах. Томск: Изд-во ТПУ, 2007. Т. 2. С. 353—355.
5. Вентцель Е. С. Теория вероятностей. М.: Академия, 2005. 576 с.

Новая книга

Свиридов А. П. Статистическая теория обучения. Монография. — М.: РГСУ, 2009. — 570 с., илл. — 131, библи. — 398 назв.

Предлагаемая монография имеет две цели: систематизированное представление результатов статистической теории обучения и подготовка специалистов по искусственному интеллекту, педагогике, психологии, информатике, обеспечению профессиональной готовности персонала, интеграции производственного и знаниевого (информационного) менеджмента.

Монография состоит из двух частей. Первая часть посвящена статистической динамике знаний (СДЗ). Предложена концепция построения и гармоничного (устойчивого) развития системы образования на основе принципов дополненности, двойственности и золотого сечения. Зона гармонии между дополненностями, противоположностями (парой полюсов) определяется в виде интервала между двумя точками золотого сечения. Предлагаются способы решения трех комплексов проблем СДЗ: точечное и интервальное оценивание количественных характеристик процессов усвоения и забывания знаний; моделирование динамики знаний и прогнозирование качества подготовки по одной или нескольким учебным дисциплинам, для чего вводятся два фундаментальных понятия: поток учебного материала по учебной дисциплине и поток усвоения (восстановления) знаний; оптимизация управления случайными процессами динамики знаний для поддержания необходимого качества профессиональной готовности.

Вторая часть посвящена рефлексии, идентификации и реализации разнообразных отношений человек—человек: педагогические, производственные отношения, отношения при психологической диагностике, определении профессиональной пригодности и др. Их общность, состоящая в оценке скрытых (латентных) качеств одного человека другим в одной из шкал измерения (чаще порядковой или интервальной) позволила рассмотреть указанные отношения на примере контроля знаний (КЗ).

Предлагаются методы анализа и синтеза одно-, двухступенчатых и последовательных планов стандартизированного КЗ, включая учет экономических показателей, нечеткие, нейросетевые и гибридные нейронечеткие модели и системы для идентификации и реализации указанных отношений. Последние модели и системы объединяют достоинства нейронных сетей (способность к обучению) и нечетких систем (наглядность представления знаний о предметной области).

Изложение сопровождается многочисленными примерами.

Монографию можно заказать: prof_sviridov@mail.ru, по тел. 8-495-336-86-73 или по адресу: Москва, ул. В. Пика, дом 4, проф. Свиридову А. П.

Т. М. Волосатова, канд. техн. наук, доц.,
Д. Е. Беломойцев, ассистент,
 МГТУ им. Н. Э. Баумана,
 e-mail: dmitry.belomoytsev@gmail.ru

Технологии и библиотеки методов построения пикосетей мобильных устройств

Выполнен анализ материалов по существующим методикам проектирования каналов передачи защищенных сообщений. Предложена модель канала беспроводного соединения мобильных устройств. Разработана программная реализация клиент-серверного взаимодействия.

Ключевые слова: проектирование, автоматизация, беспроводной, мобильный.

На современном уровне развития вычислительной техники, как правило, персональные компьютеры или рабочие станции объединяются в различного рода сети. В полной мере это относится и к мобильным устройствам. В настоящее время при выборе технического оснащения одним из решающих факторов является степень мобильности. Потребность в доступе к большим или распределенным в пространстве массивам информации определяет необходимость подключения к сетевым магистралям. При этом чрезвычайно актуальным является аспект мобильности при доступе к сети. Само понятие «мобильный» предполагает, что прибор способен выполнять свои функции, находясь в движении. Единственной возможностью взаимодействовать с другими устройствами и при этом не быть ограниченным в подвижности является использование беспроводных технологий связи. Потребность в обмене информацией между различными видами вычислительной техники достаточно высока. Способность технического обеспечения различных сфер деятельности человека выполнять поставленные задачи и при этом не быть жестко привязанным к своему местоположению (мобильность) становится определяющей. Для успешного выполнения задач мобильным электронным устройствам необходимо взаимодействовать друг с другом, в том числе обмениваться данными. Как показывает проведенный анализ имеющихся решений, задачи локального соединения могут быть обеспечены применением технологий по стандартам IEEE 802.11 и 802.15. Данные технологии беспроводной связи реализуются на основе модели канала передачи сообщений. Перечисленные выше стандарты определяют методики проектирования, целью которых являются синтез структуры и поиск оптимального набора значений параметров канала передачи сообщений.

Принято понимать под каналом передачи данных часть коммуникационной сети, состоящую из технических средств передачи и приема данных, включая линию связи, а также из средств алгоритмического обеспечения и протоколов взаимодействия, предназначенную для трансляции определяемого передаваемыми данными набора сигналов между пользователями канала. Структурой канала предусматривается влияние внутренних шумов системы на узлы передатчика и приемника, а также приложение внешних шумов в виде атак активного и пассивного типов.

Анализ принципов формирования структуры каналов в соответствии со стандартами IEEE 802.11 и 802.15, выделение решаемых с их помощью задач и перечня предотвращаемых внешних факторов позволили выявить недостатки содержащихся в стандартах методик проектирования каналов. Это неспособность обеспечить необходимую гибкость для оперативного реагирования на возникающие факторы воздействия на сообщения в каналах и, следовательно, невозможность внесения изменений в работу канала для устранения последствий внешних непредусмотренных заранее воздействий.

Для устранения отмеченных недостатков предлагается методика, основанная на блочно-иерархическом подходе к проектированию каналов, в которой учтены динамически изменяющиеся условия передачи сообщений во внешней среде. Получена клиент-серверная модель беспроводного соединения мобильных устройств. С использованием платформы J2ME предложена программная реализация, которая обеспечивает выполнение отраженных в модели функций соединения. Разработанная модель и программа позволяют описывать процесс обмена информацией, моделировать процедуры установления и передачи данных в беспроводных соединениях.

Подробное описание методики содержится в приложении к журналу «Информационные технологии» № 4 за 2006 г.

Список литературы

1. **Норенков И. П.** Системы автоматизированного проектирования: Кн.1. Принципы построения и структура. М.: Высш. шк., 1986.
2. **Норенков И. П., Трудоношин В. А.** Телекоммуникационные технологии и сети. М.: Изд-во МГТУ им. Н. Э. Баумана, 2000.
3. **Беломойцев Д. Е.** Методика проектирования процесса безопасной передачи данных в беспроводных соединениях мобильных устройств // Наука и образование. 2008. № 5. <http://technomag.edu.ru/doc/93258.html>
4. **Волосатова Т. М., Беломойцев Д. Е.** Технологии и библиотеки методов построения пикосетей мобильных устройств // Информационные технологии. Приложение к журналу. 2006. № 4. 32 с.
5. **Волосатова Т. М., Чичварин Н. В., Беломойцев Д. Е.** GPS-навигация и контроль доступа в пикосетях мобильных телефонов // Образование через науку: Тез. докл. Межд. конф. 2005. С. 48–49.

УДК 004.056.52, 004.056.53

О. О. Андреев, науч. сотр.,
Институт проблем информационной
безопасности МГУ им. М. В. Ломоносова,
e-mail: olegoandreev@yandex.ru

Интеграция моделей логического разграничения доступа, описанных на специализированном языке

Описан язык задания моделей логического разграничения доступа к ресурсам информационно-вычислительных систем. Рассмотрены вопросы автоматизированного анализа таких моделей, заданных с помощью описанного языка.

Ключевые слова: информационная безопасность, разграничение доступа, интеграция моделей разграничения доступа.

Введение

В связи с широким использованием информационно-вычислительных комплексов для решения практически значимых задач все больший интерес в последнее время проявляется к средствам описания политики их информационной безопасности и, в частности, к моделям логического разграничения доступа к их ресурсам. Доступ к подобным ресурсам, к которым относятся информационные активы, средства вычислительной техники и коммуникаций, далее для краткости изложения будем именовать просто доступом. Средства логического разграничения доступа (далее для краткости — разграничения доступа), основывающиеся на этих моделях, занимают одно из центральных мест в числе других компонентов политики безопасности, включающей также средства идентификации/аутентификации пользователей и процессов, действующих от их имени, шифрования и иные компоненты [1]. Механизмы разграничения доступа могут быть встроены как в операционные системы (ОС) и системные сервисы, так и в сервисы прикладного уровня.

Существенным недостатком является гетерогенность средств описания моделей разграничения доступа в отдельных компонентах программной системы. Механизмы разграничения доступа,

встроенные в различные компоненты, зачастую обладают собственным языком описания набора правил, в соответствии с которыми доступ будет разрешаться или запрещаться. Это обстоятельство затрудняет проверку пользователем, отвечающим за информационную безопасность системы (в дальнейшем именуемым офицером безопасности), или администратором соответствия механизмов разграничения доступа, которые реализуются в отдельных компонентах сложно организованных систем, политике безопасности системы в целом. Отмеченные выше и ряд других недостатков эксплуатирующихся в настоящее время систем стимулируют работы по созданию новых логико-языковых средств описания моделей разграничения доступа, таких как eXtended Access Control Markup Language (XACML) [2], Enterprise Privacy Authorisation Language [3].

В ходе жизненного цикла информационных систем происходят постоянные изменения, обновления, объединения и разделения, связанные с соответствующими процессами, происходящими в организациях, которым принадлежат эти системы. Эти процессы связаны с изменением структуры организаций, внутренних регламентов, стандартов и нормативно-правовых документов, регламентирующих их деятельность. Подобные изменения должны отражаться в политике информационной безопасности организаций и, в частности, в моделях логического разграничения доступа. Процесс создания и изменения моделей разграничения доступа и, в частности, их интеграции или разделения на несколько моделей является крайне трудоемким и подверженным ошибкам. Это обстоятельство объясняет важность и необходимость решения задачи автоматизированного анализа свойств моделей логического разграничения доступа.

В настоящей статье описывается разработанный автором язык описания моделей разграничения доступа и алгоритм сравнения моделей, заданных на этом языке.

Описание разработанного языка моделей разграничения доступа

В качестве основы для разработки языка описания моделей разграничения доступа, рассматриваемого в данной работе, выбран язык XACML как наиболее популярный и исследуемый в на-

стоящее время. Этот язык является декларативным, основанным на XML языком описания моделей логического разграничения доступа. Он стандартизован международной организацией OASIS (Organization for the Advancement of Structured Information Standards) [4].

Язык XACML получил распространение благодаря богатым выразительным средствам, позволяющим задавать широкий спектр моделей разграничения доступа, в их числе такие широко используемые модели, как дискреционная и многоуровневая.

Несмотря на преимущества XACML является излишне усложненным для изучения в рамках исследовательских работ. В работах такого рода широта описательных возможностей, введенная в язык для облегчения интеграции с существующими системами и механизмами, является избыточной и загромождающей исследование несущественными деталями. Кроме этого, синтаксис XACML с позиции конечного пользователя является фактически нечитаемым.

Указанные недостатки приводят к необходимости разработки отдельного языка, который должен быть в большой степени совместим с XACML по семантике, однако он должен быть проще для анализа и использования.

Как и большинство других, модели разграничения доступа, которые описываются данным языком, базируются на трех основных понятиях: *субъект*; *объект*; *доступ*. Основной задачей подсистемы разграничения доступа является выдача ответов на запросы "может ли конкретный субъект получить желаемый доступ к данному объекту".

Основным понятием, на котором базируются модели разграничения доступа, задаваемые с помощью разработанного автором языка, является понятие атрибута безопасности, в дальнейшем называемым просто атрибутом. Каждый объект и субъект могут иметь некоторые задаваемые пользователем или системой атрибуты. Доступ разрешается или запрещается в соответствии со значением, которое выдается заданной в конкретной модели булевозначной функции, зависящей от атрибутов субъекта, объекта, запрашиваемого доступа и *окружения*. Эта функция называется в дальнейшем функцией доступа. Под окружением понимается некоторый стандартный набор атрибутов, таких как "текущее время". Такие атрибуты являются общими для всех субъектов и объектов. Примерами атрибутов могут быть время последнего обращения пользователя к объекту или данные о том, в какой должности находится пользователь, каков уровень секретности у запрашиваемого объекта.

В языке, предлагаемом в данной работе, атрибуты могут быть следующих типов: булевский; це-

лочисленный; вещественный; строковый; множество значений определенного типа (одного из вышеперечисленных или типа множество).

Представление модели разграничения доступа в рассматриваемом языке структурно подразделяется на *модели* и *правила*. Правило представляет собой базовую единицу модели, состоящую из *области применения*, *условия* и *результата*.

Область применения правила состоит из четырех предикатов, описывающих допустимые субъекты, объекты, типы доступа и окружение, к которым это правило может быть применено. Условие является булевым предикатом, который определяет результат применения правила к запрашиваемому доступу. Оно зависит от атрибутов субъекта, объекта, типа доступа и окружения. Область применения и условие могут быть построены из логических связей, базовых предикатов, атрибутов субъектов, объектов, типа доступа и окружения и константных значений. Результат правила может быть либо "разрешено", либо "запрещено".

Вычисление результата применения правила к запросу на доступ выглядит следующим образом: проверяется область применения правила; в случае, если субъект, объект, тип доступа или окружение не подходят под область применения, правило считается неприменимым. В случае, если правило применимо, проверяется условие; если условие истинно, то правило возвращает свой результат, в противном случае правило возвращает отрицание результата (т. е. "запрещено" для "разрешено" и наоборот).

Модель состоит из области применения, набора правил и *алгоритма объединения правил*.

Алгоритм комбинирования задает способ, которым будут объединяться результаты применения правил к вопросу разрешения доступа. Алгоритм может быть "приоритет разрешения" или "приоритет запрещения".

Вычисление результата применения модели к запросу на доступ устроено следующим образом: проверяется область применения модели, в случае, если субъект, объект, тип доступа или окружение не подходят под область применения, модель считается неприменимой; после этого к запросу последовательно применяются все правила. Если ни одно из правил или моделей не применимо, то модель считается тоже неприменимой, если алгоритм комбинирования модели — "приоритет разрешения", то в случае существования хоть одного правила или модели, которая разрешает доступ, модель разрешает доступ, в противном случае доступ запрещается. Если же алгоритм комбинирования модели — "приоритет запрещения", то в случае существования хотя бы одного правила или модели, которая запрещает доступ,

вся модель запрещает доступ, в противном случае доступ разрешается.

Анализ моделей логического разграничения доступа

Одной из практически важных задач при анализе моделей логического разграничения доступа является проверка того факта, какая из анализируемых моделей является наиболее "сильной", т. е. наиболее ограничивающей. Необходимость в решении такой задачи может появляться в процессе обновления политики безопасности организации или в ходе интеграции политики безопасности отдельных организаций, когда требуется изменить или объединить несколько моделей логического разграничения доступа. При проведении таких изменений крайне важно удостовериться, что получающаяся модель не является более "слабой", чем предыдущие модели. Разработанный алгоритм позволяет автоматизированно проверять указанное свойство для некоторого класса моделей, записанных на предложенном языке.

Перед описанием алгоритма анализа моделей необходимо задать формальное представление моделей разграничения доступа, которые могут быть заданы с помощью предлагаемого языка.

Определение. Зададим:

- S — множество имен атрибутов субъектов;
- O — множество имен атрибутов объектов;
- E — множество имен атрибутов окружения;
- A — множество типов доступа;
- $V = Integer \cup Float \cup String \cup Set$ — множество значений атрибутов;
- F — набор унарных и бинарных функций, определенных на подмножествах V .

В заданных обозначениях одно правило модели разграничения доступа представляется в виде пары формул $(Target, Condition)$, которые составлены с помощью композиции функций из F , имен из $S \cup O \cup E \cup A$ и констант из V .

Формула $Target$ задает область применимости правила, а $Condition$ — его условие. При этом правило применимо тогда, когда значение формулы $Target$ на атрибутах субъекта, объекта и окружения истинно. Правило запрещает доступ в тех случаях, когда значение формулы $Condition$ на тех же атрибутах субъекта, объекта и окружения ложно. В противном случае доступ разрешается. Данное определение полностью соответствует неформальному определению правила, приведенному в предыдущем разделе, за исключением отсутствия результата правила, так как он считается включенным в формулу $Condition$.

При заданных определениях становится возможным определить модель разграничения досту-

па как булеву формулу разрешения доступа. Такая формула зависит от атрибутов субъекта, объекта, окружения и типа доступа и принимает истинное значение в том случае, если доступ должен быть разрешен, в противном случае — ложное.

Теорема. В том случае, если алгоритм объединения правил модели — приоритет разрешения, формула разрешения доступа выглядит как

$$Target \wedge ((Target_1 \wedge Condition_1) \vee \dots \vee (Target_n \wedge Condition_n)),$$

где $Target$ — формула для области применимости модели, а $(Target_n, Condition_n)$ — формулы представления отдельных правил модели. В том случае, если алгоритм объединения правил модели — приоритет запрещения, формула выглядит как

$$Target \wedge (Target_1 \vee \dots \vee Target_n) \wedge ((\neg Target_1 \vee Condition_1) \wedge \dots \wedge (\neg Target_n \vee Condition_n))$$

при тех же обозначениях.

Доказательство. Рассмотрим случай, когда алгоритм объединения правил модели — приоритет разрешения. В этом случае исходя из данного в предыдущем разделе определения этого алгоритма можно заключить, что модель разрешает доступ тогда и только тогда, когда доступ входит в область определения модели и существует хоть одно правило, в область определения которого входит этот доступ и которое разрешает его. Это условие эквивалентно одновременной истинности формулы $Target$ и одной из конъюнкций $Target_n \wedge Condition_n$, что и записано в условии теоремы.

В случае, когда алгоритм — приоритет запрещения, его определение эквивалентно следующему: модель запрещает доступ тогда и только тогда, когда существует правило, под которое доступ подходит и которое его запрещает, или правил, под которые доступ подходит, не существует, или доступ не подходит под область определения модели. Это условие записывается следующим образом:

$$\neg Target \vee (\neg Target_1 \wedge \dots \wedge \neg Target_n) \vee ((Target_1 \wedge \neg Condition_1) \vee \dots \vee (Target_n \wedge \neg Condition_n)).$$

Обращая его для получения условия, когда модель разрешает доступ, получаем формулу из условия теоремы. Теорема доказана.

Введение формальных определений правила и модели разграничения доступа, записанных на предложенном языке описания, делает возможным введение определения проверяемого разработанным алгоритмом критерия.

Определение. Одна модель разграничения доступа с формулой разрешения доступа F_1 называ-

ется *более слабой*, чем другая с формулой разрешения доступа F_2 в том случае, когда при любых значениях атрибутов и типе доступа истинно следствие $F_2 \rightarrow F_1$. Одна модель разграничения доступа называется более сильной, чем вторая, если вторая является более слабой, чем первая. С неформальной точки зрения более слабая модель разрешает все доступы, что и более сильная, и, возможно, еще какие-то дополнительные.

Описываемый в данной работе алгоритм позволяет проверять, является ли одна модель разграничения доступа более сильной, чем вторая, при определенных ограничениях на подаваемые ему на вход модели. Ограничение налагается на формулы разрешения доступа моделей и, как следствие, на формулы *Target* и *Condition* их правил. Оно состоит в том, что множество F функций, на основе которого строятся эти формулы, содержит только булевы связки \vee, \wedge, \neg , операции сравнения $=, \neq, <, >$, и операции включения подмножеств и элементов $\subset, \supset, \in, \notin$. Необходимо заметить, что указанные функции, за исключением булевых связок, могут иметь лишь аргументы, не являющиеся булевскими и дают булевский результат.

Определение. Назовем *литералом* атрибут, отрицание атрибута или выражение вида $(a \text{ op } b)$, где a и b — атрибуты или константы, а *op* — операция из множества $\{=, \neq, <, >, \subset, \supset, \in, \notin\}$ (т. е. множество разрешенных функций F , за исключением логических связок).

Исходя из сделанного в предыдущем абзаце замечания можно заключить, что функции из разрешенного множества F , за исключением логических связок, могут использоваться лишь в составе литералов. Таким образом, вся формула разграничения доступа может рассматриваться как булева формула относительно заданных литералов. Кроме этого, введем определение операции *отрицания литерала*. Для булевой константы, атрибута и отрицания атрибута она определяется естественным способом. Отрицанием выражения с двумя аргументами является выражение с теми же аргументами и с противоположной операцией (например, операции " $<$ " соответствует обратная " $>$ ", операции " \subset " соответствует " \supset ").

Основной алгоритм проверки моделей разграничения доступа строится на основе алгоритма проверки на противоречивость системы отношений типа $\{a_i \text{ op } b_i\}$, где a_i и b_i — атрибуты или константы, а *op* — одна из функций a множества F , за исключением булевских. Опишем вспомогательный алгоритм, при этом будем считать, что все b_i имеют один и тот же тип.

Проверка противоречивости системы отношений для случая, когда b_i имеет целочисленный, вещественный или строковый тип, проводится

одинаковым способом, так как в таких системах допустимы общие операции — это операции сравнения $=, \neq, <, >$.

Множество $\{a_i\} \cup \{b_i\}$ факторизуется по отношениям равенства, присутствующим в подсистеме, т. е. "склеиваются" равные значения. После факторизации получившееся множество проверяется на противоречие отношениям неравенства. В том случае, если отношения неравенства противоречат отношениям равенства, система является противоречивой. В противном случае строится направленный граф с вершинами из факторизованного множества. Ребра расставляются между вершинами, находящимися в отношении "меньше" или "больше", они направлены от большей вершины к меньшей. С помощью одного из существующих алгоритмов выполняется проверка на существование в получившемся графе направленных циклов. В том случае, если такие циклы существуют, подсистема является противоречивой, так как они обозначают последовательности неравенств вида $x_1 < \dots < x_n < x_1$, которые являются противоречивыми.

В случае, когда рассматриваются подсистемы отношений, соответствующих вещественным и строковым атрибутам, проверка подсистемы завершается, и подсистема считается непротиворечивой. Это связано с возможностью задания значений атрибутам, присутствующим в отношениях подсистемы, с помощью процесса, именуемого топологической сортировкой [5]. В случае рассмотрения подсистемы, соответствующей целочисленному типу, такое задание значений не всегда возможно из-за присутствия в графе констант. Например, в системе $1 < x, x < y, y < 2$ задание значений атрибутам невозможно вследствие того, что между константами 1 и 2 существует лишь одно целочисленное значение. Это условие необходимо проверять отдельно, перечисляя все возможные топологические порядки, получающиеся из этого графа, и проверяя возможность задания атрибутам значений в соответствии с этими порядками с учетом констант.

Проверка на противоречивость подсистемы отношений с множественными типами проводится в два этапа. Первый этап, включающий построение направленного графа с ребрами, соответствующими отношениям "является подмножеством" и "является надмножеством", проводится аналогично предыдущему шагу.

На втором шаге к графу добавляются вершины, соответствующие оставшимся атрибутам и константам, используемым в исследуемой подсистеме отношений, а также ребра двух новых типов. Первый тип ребер соответствует отношению "включено в множество", второй тип — "не включено в множество". После постановки этих ребер

происходит их "продвижение" по направленным путям, основанное на следующем простом правиле: если элемент включен в множество, то он включен и в его надмножество, если же элемент не включен в множество, то он не включен и в его подмножество. При продвижении ребер проверяется условие, что не образуется пара вершин, между которыми идут ребра одновременно первого и второго типа. Это условие обозначает отсутствие пары противоречивых отношений, что элемент включается и не включается в множество одновременно. Как и в случае с подсистемой отношений целочисленного типа, дополнительно проверяются все топологические порядки на предмет возможности задания значений атрибутам.

Опишем теперь последовательность шагов основного алгоритма в предположении, что на вход поданы две модели разграничения доступа, удовлетворяющие указанному ограничению, и проверяется условие, что первая модель более сильная, чем вторая. Формула разрешения доступа первой модели будет обозначаться как F_1 , формула разрешения доступа второй — как F_2 . Алгоритм должен в соответствии с определением выполнять проверку истинности формулы $F_1 \rightarrow F_2$ на всех возможных значениях аргументов, которая эквивалентна проверке ложности формулы $F_1 \wedge \neg F_2$, обозначаемой в дальнейшем F . Эта формула удовлетворяет тому же ограничению, что и формулы F_1 и F_2 .

Алгоритм:

1. Формула F рассматривается как булева формула относительно введенного определения литерала и приводится к виду дизъюнктивной нормальной формы с помощью эффективного алгоритма, такого как описан в [6].

2. Для ложности дизъюнкции каждый из дизъюнктов должен был ложным при любых значениях атрибутов, поэтому последующие шаги выполняются последовательно для всех получившихся на первом шаге дизъюнктов.

3. Каждый из дизъюнктов, ложность которого необходимо проверить, выглядит в общем виде

как $a_1 \wedge \dots \wedge a_n \wedge (a_{n+1} \text{ op}_1 b_{n+1}) \wedge \dots \wedge (a_{n+k} \text{ op}_k b_{n+k})$, где $a_1, \dots, a_{n+k}, b_1, \dots, b_{n+k}$ — атрибуты, отрицания атрибутов или константы, $a, \text{op}_1, \dots, \text{op}_k$ — операции. Для ложности такого дизъюнкта возможны два варианта: для каждого атрибута из множества $\{a_1, \dots, a_n\}$ в нем же должно присутствовать его отрицание или должна быть противоречива система отношений $(a_{n+1} \text{ op}_1 b_{n+1}) \wedge \dots \wedge (a_{n+k} \text{ op}_k b_{n+k})$. Последнее свойство проверяется с помощью представленного выше алгоритма проверки системы отношений на противоречивость для каждой из отдельных подсистем отношений, оперирующих с атрибутами целочисленного, вещественного, строкового или множественного типа. Для противоречивости всей системы отношений должна быть противоречива хотя бы одна из подсистем.

Заключение

В работе был предложен язык описания моделей разграничения доступа и представлен алгоритм сравнения пары моделей, записанных с помощью этого языка. Представленный алгоритм позволяет упрощать работу администраторов безопасности информационных систем, в которых используются средства разграничения доступа, основанные на данном языке, при изменениях моделей разграничения доступа.

Список литературы

1. **Васенин В. А.** Проблемы математического, алгоритмического и программного обеспечения компьютерной безопасности в Интернет // Матер. конф. МАБИТ-03. Москва, 2004. С. 111—143.
2. **eXtensible Access Control Markup Language (XACML) Committee Specification.** OASIS Open, 2003.
3. **Enterprise Privacy Authorization Language.** IBM Research Report, 2003.
4. **Сайт** организации OASIS, <http://www.oasis-open.org/>
5. **Кормен Т. Х., Лейзерсон Ч. И., Ривест Р. Л., Штайн К.** Алгоритмы: построение и анализ. 2-е изд. М.: Вильямс, 2006. 1296 с.
6. **Plaisted D. A. and Greenbaum S.** A structure-preserving clause form translation // Journal of Symbolic Computation. 1986. Vol. 2.

13-я специализированная выставка "Связь"

16—18 марта 2010 год

Организаторы: Министерство информатизации и связи Республики Татарстан, Министерство промышленности и торговли Республики Татарстан, Мэрия города Казань, Выставочный центр "Казанская ярмарка"

В программе выставки: тематические семинары, конференции, учебно-методические сборы, презентации проектов.

Основные тематические направления по тематике журнала:

- Системы контроля, информационная безопасность каналов связи
- Услуги сетей связи (местная, международная связь, IP-телефония, доступ в Интернет)
- Локальные, корпоративные и глобальные сети (оборудование, технологии)
- Видео, конференцсвязь
- Перспективные разработки в области связи

Тел./факс: (843) 570-51-07, 570-51-11 — круглосуточно
Россия 420059, г. Казань, Оренбургский тракт, 8, ОАО "Казанская ярмарка"
E-mail: d4@expokazan.ru, <http://www.exposvyaz.ru>, www.expokazan.ru

О. С. Ушмаев, канд. техн. наук. ст. науч. сотр.,
Институт проблем информатики РАН,
ООО "Биолинк Солюшенс", директор по науке,
e-mail: oushmaev@ipiran.ru

Проблемы применения биометрической идентификации в комплексной защите информации

Рассмотрена проблема применения биометрии в комплексной защите информации. Дан общий обзор биометрических методов и их преимуществ перед парольной защитой. Проанализированы типовые внедрения биометрии по направлениям обеспечения информационной безопасности: контроль физического доступа, идентификация в сетях, защита данных. Выработан подход к комплексированию перечисленных направлений на базе биометрического идентификатора.

Ключевые слова: биометрическая идентификация, идентификация пользователя, контроль доступа.

Введение

В настоящее время активно совершенствуются различные механизмы обеспечения информационной безопасности в корпоративных, ведомственных и домашних сетях. Особенное внимание уделяется предотвращению несанкционированного доступа к информационным ресурсам. Несмотря на значительный прогресс в этой области, наиболее слабым звеном в любой системе остается человеческий фактор [1—6], а именно, надежная идентификация пользователя.

Если рассматривать проблему идентификации личности в целом, то можно выделить три направления: контроль физического доступа к аппаратным средствам; идентификация пользователя в сети и прикладных программах; защита данных. Для всех перечисленных направлений используются следующие идентификаторы:

- владение информацией (пароль, PIN-код, условная фраза, ответы на вопросы и т. д.);
- владение вещью (токен, смарт-карта, ключ и т. д.);
- биометрия, т. е. характеристики, основанные на физических или поведенческих особенностях конкретного человека.

Каждый из перечисленных вариантов имеет свои преимущества и недостатки. Отметим отдельные из них. Идентификация по парольной информации приводит к таким проблемам, как потеря пароля, слабая стойкость к взлому, использование чужих паролей при несанкционирован-

ном доступе к информационным ресурсам. Наиболее уязвимым местом ключей и смарт-карт является риск их физической утраты. В большинстве случаев восстановление паролей и ключей приводит к увеличению затрат. Так, по данным *Gartner* восстановление утерянного пароля обходится в сумму от 14 до 28 долл. США. В некоторых случаях утрата идентификатора приводит к потере ценной информации (например в случае криптозащиты данных). Ущерб от таких событий не поддаются оценке, поскольку в большинстве случаев информация о них недоступна для аудиторов и исследователей.

Биометрия лишена перечисленных недостатков — ее нельзя утерять, забыть, подменить и пр. В то же время биометрия потенциально не обладает ничем не ограниченным разнообразием, которое дает пароль (число комбинаций растет экспоненциально с каждым добавленным символом). В то же время опыт реализации высокопроизводительных идентификационных систем [7—15] показывает, что разнообразия биометрических образцов достаточно для решения большинства прикладных задач, связанных с проблемами информационной безопасности.

Вопросы использования паролей и ключей в достаточной степени проработаны. Биометрия в защите информации является относительно новым направлением. Основной целью данной статьи является систематизация подходов к использованию биометрии в защите информации. Далее статья организована следующим образом. В разделе 1 дана характеристика методов биометрической идентификации. Раздел 2 посвящен исследованию специфики использования биометрической идентификации в задачах защиты информации. В разделе 3 даны рекомендации по комплексному внедрению биометрии. В заключении представлены основные выводы.

1. Биометрическая идентификация

По данным компании *Acuity Market Intelligence* рынок биометрических систем в 2008 г. распределился следующим образом (рис. 1, см. третью сторону обложки). Доля отпечатков пальцев традиционно составила более 50 %. Вторым по распространенности методом является изображение лица. Радужная оболочка (7 %) занимает третье место в монетарном исчислении. В единицах продукции доля меньше, поскольку радужная оболочка является дорогой биометрией, особенно на фоне отпечатков пальцев и изображения лица.

Каждый из упомянутых методов биометрической идентификации имеет свои положительные и отри-

Характеристика методов биометрической идентификации

Метод	Средняя стоимость терминального устройства, тыс. руб.	Ошибка верификации 2-го рода, %	Положительные особенности	Отрицательные особенности
Отпечатки пальцев	3	0,001	Доказанная устойчивость параметра к возрастным изменениям Высокое качество идентификации Компактный сканер	Ассоциация с криминалистическими системами Контактная биометрия
Изображение лица	3	1	Возможность непрерывного мониторинга, система видеонаблюдения Бесконтактная биометрия Возможность верификации человеком результатов автоматической идентификации	Большие ошибки идентификации Сильная зависимость от внешних условий: освещенность, ракурс и пр.
Радужная оболочка глаза	15	0,0001	Бесконтактная биометрия Потенциально высокая надежность Новизна биометрии — отсутствие негативных ассоциаций	Недоказанность низких ошибок идентификации Действующие "зонтичные" патенты
Геометрия руки	16	0,2	Простота технологии	Временная неустойчивость Высокие ошибки Контактная биометрия
Голос	2	1	Бесконтактная биометрия "Бесконечное" разнообразие парольных фраз	Шумная технология Слабая устойчивость к простым атакам (воспроизведение записи на диктофон)
Сетчатка глаза	100	0	Крайне высокое качество идентификации	Крайне высокая стоимость Недоказанная безвредность способа сканирования

цательные особенности. Сводная характеристика различных методов представлена в табл. 1 [10, 17].

Помимо ошибок распознавания, стоимости и отличительных особенностей, важной характеристикой биометрических методов являются их эксплуатационные показатели (время обработки, объем данных, время идентификации). Перед анализом указанных показателей поясним принципы функционирования биометрических систем.

Согласно сложившейся отраслевой практике и действующим стандартам, биометрические системы функционируют в следующих режимах:

1. Регистрация (*enroll*). При регистрации пользователь заносит свои персональные данные и предъявляет биометрический образец (например, оставляет отпечаток пальца или делает цифровую фотографию). В обмен получает идентификационную информацию. В системе фиксируется предъявленный биометрический образец и внесенные персональные данные.

2. Верификация (*verification*). Человек предъявляет идентификационную информацию (это может быть паспорт, токен, имя пользователя) и биометрический образец. Система сравнивает предъявленный и хранимый биометрические образцы и дает ответ тот ли это человек, за кого он себя выдает.

3. Идентификация (*identification*). Системе предъявляется биометрический образец. Система отвечает на вопрос, принадлежит ли образец пользователю системы. Если да, то кому.

Схематично режимы работы биометрической системы изображены на рис. 2 (см. третью сторону обложки). Как видно, логика реализации биометри-

ческой системы не зависит от того, используется ли для идентификации отпечаток пальца, форма лица или другая биометрическая информация. Это соображение стало основой важных отраслевых стандартов. В частности, bioAPI (в российском гармонизированном варианте — ГОСТ Р ИСО/МЭК 19784).

Как видно из рис. 2 (см. третью сторону обложки), при технической оценке биометрической идентификации пользователя важны следующие параметры:

- размер биометрического образца;
- размер биометрического шаблона;

Таблица 2

Средние эксплуатационные показатели биометрических технологий [10, 17]

Технология	Время верификации ¹ , сравнений/с	Размер шаблона, байт	Размер образца, байт	Время создания шаблона ¹ , с
Отпечатки пальцев				
BioLink ATK	9 000	600—3000	100 К	0,2
Neurotechnology MegaMatcher	10 000	600—3000		0,3
Изображение лица				
Neurotechnology VeriLook	60 000	3000	200 К	0,6
Радужная оболочка глаза				
Biolink Idenium	200 000	3400	150 К	0,1
Голос				
ЦРТ Трал	1 с	54 К	N/A	5

¹ В расчете на одно ядро тактовой частотой 1 ГГц.

- время создания шаблона;
- время верификации (время идентификации обычно равняется произведению объема базы на время верификации).

Сводные характеристики производительности по различным биометрическим методам приведены в табл. 2.

Как видно из приведенных таблиц, наименее экономной по времени и объемам информации является голосовая биометрия. Наиболее производительной — изображение лица. Наиболее удобной — радужная оболочка глаза. Самой надежной — отпечатки пальцев.

2. Применение биометрической идентификации в защите информации

Рассмотрим направления защиты информации: физический доступ, идентификация/аутентификация пользователя в ОС и защита данных, по отдельности.

Основными решениями в направлении физического доступа к аппаратным средствам являются системы контроля и управления доступом (СКУД). Работа биометрической СКУД выстраивается в полном соответствии с рис. 2. В случае, если сотруднику дополнительно выдается токен или смарт-карта, система преимущественно функционирует в режиме верификации. Если смарт-карта не выдается, то решение о допуске принимается на основе результатов идентификации.

Специфической особенностью СКУД является то, что сканер устанавливается на вычислительных средствах малой производительности (тактовая частота центрального процессора составляет порядка десятков мегагерц), что накладывает дополнительные ограничения на используемые технологии. В некоторых случаях снижения загрузки вычислителя на борту терминального сканера можно добиться перераспределением части вычислений на сервер (рис. 3, см. четвертую сторону обложки).

В таком случае основным ограничением, которое накладывается на терминал, является возможность создания шаблона за приемлемое время (в пределах 2 с). Передача исходного биометрического образца по сети, даже по защищенным протоколам, является недопустимой с точки зрения информационной безопасности. Некоторые СКУД допускают использование изолированных терминалов. Тогда суммарное время t идентификации определяется как

$$t = t_e + Nt_m,$$

где t_e — время создания шаблона; t_m — время сравнения один к одному, N — размер базы.

При вычислении времени идентификации следует учитывать, что производительность процес-

сора на терминальном устройстве составляет от 1 до 5 % вычислительной мощности среднего ПЭВМ. Для уменьшения времени идентификации в некоторых случаях используют дополнительный PIN-код, который позволяет разделить всю совокупность зарегистрированных пользователей на более мелкие группы.

С учетом указанного ограничения на производительность терминальных устройств в СКУД используются следующие технологии: отпечаток пальца, радужная оболочка глаза, форма лица, геометрия ладони.

При идентификации в ОС и прикладных программах проблема слабой производительности СКУД неактуальна. Здесь наиболее важным фактором становится стоимость сканера (так как рабочих мест обычно значительно больше точек контроля СКУД) и надежность (чтобы гарантированно отказаться от предъявления дополнительной информации в виде имени пользователя или смарт-карты). С точки зрения реализации наиболее перспективным вариантом является интеграция биометрии в штатные механизмы ОС. В случае ОС *MS Windows* такие реализации возможны на базе *MS Active Directory*. Тогда внедрение биометрии, по-крупному, предполагает замену штатных механизмов сравнения хэшированных паролей на сервис сравнения биометрических шаблонов (рис. 4, см. четвертую сторону обложки).

В системах биометрической идентификации пользователя в сетях и прикладных программах наиболее перспективными технологиями являются отпечатки пальцев (недорогое решение), радужная оболочка глаза и форма лица (решение для небольших коллективов). Однако при внедрении биометрической идентификации в крупные распределенные сети могут возникать дополнительные проблемы с качеством идентификации. В таком случае одним из наиболее перспективных направлений является мультибиометрическая идентификация, использующая одновременно несколько биометрических параметров. Систематические решения по разработке мультибиометрических технологий изложены в работах [7—9, 18].

При более глубокой проработке политики информационной безопасности на базе биометрической идентификации можно привлечь биометрию к решению задачи генерации ключей к криптографическим приложениям. Единственным требованием к биометрии является возможность создания устойчивых ключей достаточной длины. На сегодняшний день известны попытки создать системы криптозащиты с использованием отпечатков пальцев, радужной оболочки глаза и голоса. Наименее уязвимой к простым атакам является радужная оболочка глаза. Отпечатки пальцев оставляют следы на сканерах, которые потенциально

Таблица 3

**Применение методов биометрической идентификации
в защите информации**

Биометрия	Физический доступ	Аутентификация в ОС	Защита данных
Отпечаток пальца	+	+	+
Радужная оболочка	+	+	+
Изображение лица	+	+	
Голос	+		
Геометрия ладони	+		+

могут быть повторно использованы. Данная проблема малоактуальна в СКУД, где следы представляют смесь множества приложений, но становится важной в критичных криптографических приложениях. Однако технические средства позволяют создавать покрытия сканеров, которые близки к устранению данной проблемы. Голос минимально устойчив даже к простейшим *replay*-атакам.

3. Биометрическая идентификация в комплексной защите информации

Если рассматривать проблему идентификации личности в системе информационной безопасности в целом, то можно попытаться получить дополнительный эффект за счет внедрения биометрии одновременно на всех уровнях. Тогда первой задачей является выбор биометрии. В табл. 3 приведены сводные данные о применимости отдель-

ных биометрических методов по направлениям защиты информации.

При комплексировании отдельных направлений в единую систему информационной безопасности возможны два принципиально различных подхода к выбору используемой биометрической технологии. Первый заключается в использовании для каждого решения своего вида биометрии. Второй предполагает сквозную систему на базе одного идентификатора. Как видно из табл. 3, наиболее важными видами биометрии являются отпечатки пальцев и радужная оболочка глаза, поскольку применимы во всех перечисленных направлениях обеспечения информационной безопасности: СКУД, идентификация в ОС, шифрование данных.

При комплексном подходе к биометрической идентификации штатное функционирование укладывается в следующую последовательность действий (рис. 5, см. четвертую сторону обложки): вход в помещение, рабочая сессия, выход из помещения.

Мероприятия, проводимые на каждом этапе, приведены в табл. 4. При этом конкретные действия системы безопасности могут регламентироваться внутренними корпоративными или ведомственными инструкциями.

Заключение

Биометрические технологии позволяют перейти к новому уровню в обеспечении информаци-

Таблица 4

Мероприятия по биометрической идентификации пользователя

Действие пользователя	Действия системы идентификации личности	Действия системы безопасности
Начало работы сессии		
Идентификация при входе в помещение (включении компьютера)	Верификация пользователя	Проверка прав доступа в помещение
	Протоколирование в системе учета рабочего времени и журнале системы безопасности	Проверка ограничений прав доступа (сотрудник в отпуске, нерабочее время и т. д.)
	Разблокирование учетной записи	
Рабочая сессия		
Идентификация при входе в операционную систему	Идентификация пользователя	
	Запрос к системе учета рабочего времени	Проверка факта физического входа в помещение
	Запрос к системе безопасности	Получение контекста и полномочий пользователя
Идентификация в прикладных программах	Идентификация пользователя	
Шифрование защищенных данных	Создание ключей из биометрических данных	
Окончание рабочей сессии		
Идентификация на выходе из помещения	Верификация пользователя	Проверка фактов НСД и режима безопасности, в экстренных случаях блокировка выхода
	Протоколирование в системе учета рабочего времени и журнале системы безопасности	
	Блокировка учетной записи	Блокировка действий от имени пользователя

онной безопасности. В частности, предложенный подход позволяет:

- минимизировать риск утери идентификатора и сопряженные с этим затраты;
- усилить систему информационной безопасности;
- выстроить систему информационной безопасности на базе единого идентификатора.

Проведенный анализ позволяет сделать заключение о пригодности того или иного биометрического идентификатора. В частности, наиболее перспективными технологиями в системе идентификации пользователя являются отпечатки пальцев и радужная оболочка глаза. Отдельного внимания требуют методы мультибиометрической идентификации.

Список литературы

1. Beardsley, Charles T. Is your computer insecure? // IEEE Spectrum. 1972, Jan. P. 67–78.
2. Соколов И. А., Будзко В. И., Сеницын И. Н. Построение информационно-телекоммуникационных систем высокой доступности // Научные технологии. 2005. № 6. Т. 6. Системы высокой доступности. 2005. № 1. Т. 1. С. 6–14.
3. Fejfar A. Combining Techniques for Improve Security in Automated Entry Control // Proc. of Carnahan Conf. On Crime Countermeasures. Mitre Corp. MTP-191. May 1978.
4. Соколов А. В., Шаньгин В. Ф. Защита информации в распределенных корпоративных сетях и системах. М.: ДМК, 2002. 656 с.
5. Петров А. А. Компьютерная безопасность. Криптографические методы защиты. М.: ДМК, 2000. 448 с.
6. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных: учеб. пособие для вузов / Белкин П. Ю., Михальский О. О., Першаков А. С. и др. — М.: Радио и связь, 1999. 168 с.

7. Ushmaev O. S., Novikov S. O. Integral criteria for large-scale multiple fingerprint solutions // Proc. of SPIE. Vol. 5404. Biometric Technology for Human Identification / Ed. by Anil K. Jain, Nalini K. Ratha. (SPIE, Bellingham, WA, 2004). P. 534–543.

8. Сеницын И. Н., Новиков С. О., Урмаев О. С. Развитие технологий интеграции биометрической информации // Системы и средства информатики. 2004. Вып. 14. С. 5–36.

9. Урмаев О. С., Сеницын И. Н. Опыт проектирования многофакторных биометрических систем // Труды VIII Международной научно-технической конференции. "Кибернетика и высокие технологии XXI века". Т. 1. С. 17–28.

10. Урмаев О. С., Босов А. В. Реализация концепции многофакторной биометрической идентификации в интегрированных аналитических системах // Бизнес и безопасность в России. Январь 2008. № 49. С. 104–105.

11. Урмаев О. С. Применение биометрии в аэропортах // Proc. of Biometrics TTS 2007. 22 ноября 2007 г. URL: http://www.dancom.ru/rus/AIA/Archive/RUVII_BioLinkSolutions_BiometricsInAirports.pdf

12. Урмаев О. С. Реализации концепции многофакторной биометрической идентификации в правоохранительных системах. Интерполитех-2007. http://www.dancom.ru/rus/AIA/Archive/RUVI_BioLinkSolutions_MultimodalBiometricsConcept.pdf

13. Сеницын И. Н., Губин А. В., Урмаев О. С. Метрологические и биометрические технологии и системы // История науки и техники. 2008. № 7. С. 41–44.

14. Jain A. K., Hong L., Bolle R. On-Line Fingerprint Verification // IEEE Trans. On Pattern Analysis and Machine Intelligence. 1997. 19 (4). P. 302–314.

15. Jain A. K., Hong L., Pankanti S., Bolle R. An Identity-Authentication System Using Fingerprints // Proc. of IEEE. 1997. 85 (9). P. 1365–1388.

16. Daugman J. The Importance of Being Random // Pattern Recognition. 2003. Vol. 36. N 2.

17. Урмаев О. С., Босов А. В. Реализация концепции многофакторной биометрической идентификации в интегрированных аналитических системах // Системы высокой доступности. 2007. Т. 3. № 4. С. 13–23.

18. Урмаев О. С. Сервисно-ориентированный подход к разработке мультибиометрических технологий // Информатика и ее применения. 2008. Т. 2. Вып. 3. С. 41–53.

WEB-ТЕХНОЛОГИИ

УДК 004.91

А. А. Владыкин, аспирант,

А. А. Шальто, д-р техн. наук, проф., зав. каф.,

Санкт-Петербургский государственный университет информационных технологий, механики и оптики, e-mail: shalyto@mail.ifmo.ru

Непроцедурный текстовый язык описания автоматных обработчиков XML-документов и его применение

Предложен непроцедурный язык для описания обработчиков XML-документов, который целесообразно применять для разбора этих документов. Обработчики реализованы на основе конечных автоматов, что упрощает их структуру и понимание.

Ключевые слова: XML, SAX-обработчики, автоматы.

В настоящее время язык XML [1] широко используется для разнообразных целей: от хранения настроек программ до описания векторных изображений. Поэтому все больше программ требуют поддержки этого языка, а программистам все чаще приходится писать код, который считывает

XML-документ и выделяет из него фрагменты, представляющие тот или иной интерес.

Нередко встречаются XML-документы размером в десятки и сотни гигабайт. Один из таких документов — полная база данных Википедии, доступная для скачивания в виде XML-файла [2].

Размер этого файла, включающего все статьи Википедии с историей их изменений, в сжатом архиватором bzip2 виде составляет 148 Гбайт.

Как прочитать такой документ и извлечь из него полезную информацию, например число перекрестных ссылок между статьями? Единственным вариантом в данном случае является использование технологии *Simple API for XML (SAX)* [3] и написание соответствующего обработчика, который позволит решить поставленную задачу. Известны также и другие технологии доступа к XML-данным, например *Document Object Model (DOM)* [3] или *Java API for XML Binding (JAXB)* [4], но они требуют загрузки всего документа в оперативную память. Это при таких размерах документов не представляется возможным.

Недостатком технологии SAX, затрудняющим ее использование, является сложность написания кода для реализации указанных обработчиков. В данной работе предложена технология, позволяющая значительно упростить написание обработчиков этого класса. При этом предложен не-процедурный язык описания поведения SAX-обработчика. Обработчик реализуется на основе автоматического подхода, его код генерируется автоматически.

Технология Simple API for XML

SAX — это стандартный способ разбора XML, при котором XML-разборщик (*XML-parser*) последовательно читает XML-документ и уведомляет пользовательский код (SAX-обработчик, *SAX handler*) о каждом встречающемся открывающем или закрывающем теге и о текстовом контенте между ними.

Стандартный SAX-обработчик на языке Java расширяет класс `DefaultHandler` и переопределяет следующие три метода:

```
void startElement(String uri, String localName, String qName, Attributes attrs) throws SAXException;
```

```
void endElement(String uri, String localName, String qName) throws SAXException;
```

```
void characters(char[] ch, int start, int length) throws SAXException.
```

Метод `startElement` вызывается разборщиком, когда тот обнаруживает открывающий тег, а метод `endElement` — в ответ на закрывающий тег, и, наконец, метод `characters` — для текстового контента между тегами.

SAX-обработчик как автоматизированный объект управления

Попытаемся разобраться в причинах сложности SAX-обработчиков и устранить эти причины. Этой цели можно добиться с использованием автоматического подхода [5].

SAX-обработчик получает уведомления от XML-разборщика и выполняет всю работу по преобразованию потока событий в понятные приложению команды и/или структуры данных. Он должен отслеживать текущее положение разборщика в структуре XML-документа и в зависимости от этого положения по-разному реагировать на поступающие события.

SAX-обработчик является сущностью со сложным поведением [5], так как для него характерна зависимость от предыстории. При этом SAX-обработчик можно разделить на две части:

- **управляющая часть** (автомат или система автоматов), которая отслеживает текущее положение разборщика в структуре XML-документа и выбирает соответствующие выходные воздействия на объект управления;
- **объект управления** — преобразователь потока воздействий со стороны управляющей системы в понятные приложению команды и/или структуры данных.

Наибольшую сложность обычно составляет управляющая (поведенческая) часть обработчика, поскольку отслеживание текущего положения разборщика в структуре XML-документа при традиционном подходе к программированию требует введения и учета многочисленных флагов либо других хитростей. При этом объект управления, несмотря на относительную простоту, теряется в управляющем коде.

В соответствии с парадигмой автоматического программирования, сущность со сложным поведением следует представлять в виде автоматизированного объекта управления [5]. Отделив объект управления (управляемую часть) от системы управления (управляющей части), можно значительно повысить качество кода XML-обработчика. Более того, можно автоматически генерировать управляющую часть на основе его описания на предлагаемом не-процедурном текстовом языке. Это делает написание SAX-обработчиков значительно более простым, а структуру более наглядной.

Текстовый язык описания SAX-обработчика

Рассмотрим задачу извлечения списка персоналий из таблицы, находящейся в XHTML-документе. XHTML — это HTML, соответствующий всем правилам XML. Поэтому для разбора XHTML-документа можно применить технологию SAX. Эта задача выбрана в качестве примера, поскольку на ней можно продемонстрировать сложность написания SAX-обработчика при традиционном подходе к программированию.

Алгоритм извлечения списка персоналий следующий:

- найти таблицу (`<table>`) с атрибутом `class = "header"`;
- найти таблицу без атрибута `class`;

- в таблице без атрибута пропустить первую строку (<tr>) и все строки, у которых первая ячейка имеет атрибут class = "banner";
- из оставшихся строк взять вторую и четвертую ячейки (<td>) как фамилию и имя соответственно.

В соответствии с предлагаемым подходом опишем SAX-обработчик на непроцедурном текстовом языке, специально разработанном для этой цели.

Язык этого описания состоит из следующих элементов:

- *открывающий и закрывающий теги*, являющиеся для обработчика входными воздействиями; открывающий тег может иметь ограничения на значения атрибутов (например, <td class=="banner">), выходное воздействие задается после тега в фигурных скобках (<td {object.newPerson(); });
- *скобки, используемые для группировки тегов*; со скобками могут использоваться следующие операции:
 - альтернатива (...|...);
 - повторение (...)*
 - опциональность (...)?

Грамматика этого языка приведена в *Приложении 1*.

Описание на предлагаемом языке SAX-обработчика, реализующего приведенный выше алгоритм, имеет вид:

```
<table class == "header"> </table>
<table class == null>
<tr> </tr>
(
  <tr>
    (
      <td class == "banner"> </td> |
      <td> {object.newPerson();} </td>
      <td> {capture();} </td> {object.setLastName
        (captured());}
      <td> </td>
      <td> {capture();} </td> {object.setFirstName
        (captured());}
    )
  </tr>
)*
</table>
```

В соответствии с автоматной парадигмой обработчик будет состоять из двух частей — объекта управления и автомата управления.

Объект управления предоставляет управляющей системе некоторый интерфейс. В этом примере — функции newPerson (), setLastName () и setFirstName (). Эти функции программист пишет *вручную*.

Автомат управления вызывает функции объекта управления в соответствии с выходными воздействиями, заданными в описании SAX-обработчика. При этом могут использоваться два вспомогательных метода, предоставляемых авто-

матом управления: capture () и captured (). Метод capture () инициирует захват поступающего текстового контента во временный буфер автомата. Метод captured () возвращает захваченный с предыдущего вызова capture () контент и очищает буфер.

Алгоритм построения автомата управления

Генерация управляющей части SAX-обработчика требует выделения управляющих состояний и построения переходов между ними.

Выделение состояний является достаточно простой задачей, так как состояния уже неявно выделены при составлении текстового описания XML-обработчика. Для приведенного выше примера состояния выделяются следующим образом (программа, решающая, в частности, и эту задачу, приведена в *Приложении 2*):

```
0 <table class == "header"> 2 </table> 3
<table class == null> 4
<tr> 5 </tr> 6
(
  <tr> 8
    (
      <td class == "banner"> 10 </td> |
      <td> 11 </td> 12
      <td> 13 </td> 14
      <td> 15 </td> 16
      <td> 17 </td>
    ) 9
  </tr>
)* 7
</table> 1
```

Неформально говоря, каждый открывающий или закрывающий тег переводит обработчик в новое состояние. Единственное исключение — тег в конце группы, заключенной в скобки. В этом случае новое состояние добавляется не после данного тега, а "выносится за скобки" группы, например состояние 9.

Построение переходов также не представляет большого труда. Рассмотрим переходы для некоторых состояний из примера:

- из состояния 0 существует только один переход: при получении открывающего тега table с атрибутом class, равным header, обработчик переходит в состояние 2;
- из состояния 8 возможны два перехода: в состояние 10 — при получении открывающего тега td с атрибутом class = "banner" и в состояние 11 — при получении любого тега td, который не удовлетворяет этому условию;
- из состояния 6 происходит переход в состояние 8 по открывающему тегу tr и в состояние 1 по закрывающему тегу table;
- переходы состояния 7 идентичны переходам состояния 6;
- переходы в состояния 11, 13, 14, 17 и 9 сопровождаются выходными воздействиями.

Воздействия, не предусмотренные описанием SAX-обработчика, игнорируются и не изменяют его состояние. Например, обнаружение тегов th в состоянии 5 вполне закономерно, и не должно влиять на поведение обработчика.

Программа построения управляющего автомата в целом приведена в *Приложении 2*.

Автоматическая генерация кода автомата SAX-обработчика

Имея описание управляющего автомата в виде множества состояний и переходов между ними, можно выполнить автоматическую генерацию кода автомата SAX-обработчика для произвольного языка программирования. В рамках данной работы была реализована кодогенерация автомата SAX-обработчика на языке Java.

Результатом кодогенерации является класс, который расширяет `DefaultHandler` и реализует указанные выше методы `startElement`, `endElement` и `characters`. При этом все переходы автомата по открывающим тегам размещаются в методе `startElement`, а по закрывающим — в методе `endElement`. Автомат имеет всего две внутренние переменные: текущее состояние и буфер для временного хранения текстового контента. Таким образом, расход памяти минимален.

Пример автоматически сгенерированного автомата приведен в *Приложении 3*.

В *Приложении 4* приведена структура программы SAX-обработчика.

Заключение

В связи с широким распространением языка XML задача разбора и извлечения данных из

XML-документов стала рутинной. Наиболее эффективной с точки зрения использования ресурсов компьютера технологией чтения XML-документов является SAX.

В данной работе предложен непроецедурный текстовый язык описания SAX-обработчиков и система автоматической генерации кода на его основе. Входными данными для системы является простое и наглядное текстовое описание обработчика. На основе этого описания строится управляющий автомат обработчика, который затем преобразуется в код на том или ином языке программирования. На настоящий момент реализовано преобразование автомата в код на языке Java. Остальной код обработчика, соответствующий объекту управления, пишется вручную. Предложенный подход отличается одновременно простотой написания кода и эффективностью использования ресурсов компьютера.

Описанный подход был использован при разработке приложения, с помощью которого извлечены данные из тысяч HTML-страниц, содержащих несколько типов таблиц сложной структуры.

Список литературы

1. **Official website for SAX**. URL: <http://www.saxproject.org/>
2. **Wikipedia Database Download**. URL: http://en.wikipedia.org/wiki/Wikipedia_database
3. **W3C Document Object Model**. URL: <http://www.w3.org/DOM/>
4. **JAXB Reference Implementation**. URL: <http://jaxb.dev.java.net/>
5. **Поликарпова Н. И., Шальто А. А.** Автоматное программирование. СПб.: Питер, 2009.

Приложение 1. Грамматика описания SAX-обработчика

```
S :: START_TAG | END_TAG | GROUP

START_TAG :: "<" ID OR_EXPR? ">" ACTION?

END_TAG :: "<" ID ">" ACTION?

GROUP :: "(" ( START_TAG | END_TAG | GROUP ) * ( "*" | "?" )? ")"

ACTION :: "{" CODE "}"

OR_EXPR :: AND_EXPR ( "|" AND_EXPR ) *

AND_EXPR :: TERM ( "&" TERM ) *

TERM :: ID "==" STRING | ID "!=" STRING | ID "=~" STRING
        | ID "!~" STRING | "(" OR_EXPR ")"
```

В этом описании ID — любое корректное имя тега или атрибута; STRING — произвольный текст, заключенный в кавычки, или ключевое слово `null`; CODE — произвольный на целевом языке программирования.

Приложение 2. Программа построения управляющего автомата для SAX-обработчика по его описанию на предлагаемом языке

// Реализация используемых в программе классов для экономии места не приводится

```
import java.util.List;

public class StateGenerator {

    /**
     * @param input описание обработчика в виде последовательности элементов:
     *             * открывающих и закрывающих тегов и групп
     * @return сгенерированная таблица переходов
     */
    public static TransitionTable generate(List<InputElement> input) {
        TransitionTable table = new TransitionTable();
        generate(input, table, new StateSet(0), table.addState());
        return table;
    }

    private static StateSet generate(List<InputElement> input,
        TransitionTable table, StateSet initState, int finalState) {
        StateSet states = initState.clone();
        for (int i = 0; i < input.size(); ++i) {
            InputElement elem = input.get(i);
            int localFinalState = (i + 1 == input.size()) ?
                finalState : table.addState();
            switch (elem.getType()) {
                case START_TAG:
                case END_TAG: {
                    for (Integer source : states.getAll()) {
                        table.addTransition(source, (Tag) elem, localFinalState);
                    }
                    states.setAll(localFinalState);
                    break;
                }
                case GROUP: {
                    Group group = (Group) elem;
                    StateSet finalStates = generate(
                        group, table, states, localFinalState);
                    switch (group.getQuantifier()) {
                        case NONE:
                        case QUESTION:
                            states = finalStates;
                            break;
                        case ASTERISK:
                            table.copyTransitions(
                                states.getPrimary(), localFinalState);
                            states.add(finalStates, localFinalState);
                            break;
                    }
                    break;
                }
                default:
                    throw new IllegalArgumentException();
            }
        }
        return states;
    }

    private static StateSet generate(Group group, TransitionTable table,
        StateSet initState, int finalState) {
        StateSet states = null;
        for (int j = 0; j < group.getAlternativeCount(); ++j) {
            StateSet finalStates = generate(
                group.getAlternative(j), table, initState, finalState);
            if (states == null) {
                states = finalStates;
            } else {
                states.add(finalStates);
            }
        }
        return states;
    }
}
```

Приложение 3. Код автомата SAX-обработчика, сгенерированного программой из предыдущего приложения

```
public class SAXAutomaton extends org.xml.sax.helpers.DefaultHandler {

    private int state;
    private StringBuilder buf;
    private ControlObject object;

    public SAXAutomaton(ControlObject object) {
        this.object = object;
    }

    public void startElement(String uri, String localName,
        String qName, Attributes attrs) throws SAXException {
        switch (state) {
            case 0:
                if ("table".equals(qName) && "header".equals(attrs.getValue("class"))) {
                    state = 2;
                }
                break;
            case 3:
                if ("table".equals(qName) && null == attrs.getValue("class")) {
                    state = 4;
                }
                break;
            case 4:
                if ("tr".equals(qName)) {
                    state = 5;
                }
                break;
            case 6:
                if ("tr".equals(qName)) {
                    state = 8;
                }
                break;
            case 7:
                if ("tr".equals(qName)) {
                    state = 8;
                }
                break;
            case 8:
                if ("td".equals(qName) && "banner".equals(attrs.getValue("class"))) {
                    state = 10;
                } else
                if ("td".equals(qName)) {
                    object.newPerson();
                    state = 11;
                }
                break;
            case 12:
                if ("td".equals(qName)) {
                    capture();
                    state = 13;
                }
                break;
            case 14:
                if ("td".equals(qName)) {
                    state = 15;
                }
                break;
            case 16:
                if ("td".equals(qName)) {
                    capture();
                }
            }
        }
    }
}
```

```

        state = 17;
    }
    break;
}
}

public void endElement(String uri, String localName,
    String qName) throws SAXException {
    switch (state) {
        case 2:
            if ("table".equals(qName)) {
                state = 3;
            }
            break;
        case 5:
            if ("tr".equals(qName)) {
                state = 6;
            }
            break;
        case 6:
            if ("table".equals(qName)) {
                state = 1;
            }
            break;
        case 7:
            if ("table".equals(qName)) {
                state = 1;
            }
            break;
        case 9:
            if ("tr".equals(qName)) {
                state = 7;
            }
            break;
        case 10:
            if ("td".equals(qName)) {
                state = 9;
            }
            break;
        case 11:
            if ("td".equals(qName)) {
                state = 12;
            }
            break;
        case 13:
            if ("td".equals(qName)) {
                object.setLastName(captured());
                state = 14;
            }
            break;
        case 15:
            if ("td".equals(qName)) {
                state = 16;
            }
            break;
        case 17:
            if ("td".equals(qName)) {
                object.setFirstName(captured());
                state = 9;
            }
            break;
    }
}

public void characters(char[] ch, int start, int length) {
    if (buf != null) {
        buf.append(ch, start, length);
    }
}
}

```

```

    }
}

private void capture() {
    buf = new StringBuilder();
}

private String captured() {
    String result = buf.toString();
    buf = null;
    return result;
}
}

```

Приложение 4. SAX-обработчик

```

class ControlObject {
    public void newPerson() { /* реализация */ }
    public void setLastName(String name) { /* реализация */ }
    public void setFirstName(String name) { /* реализация */ }
}

class SAXHandler {
    public static void main(String[] args) throws Exception {
        SAXParser parser = SAXParserFactory.newInstance().newSAXParser();
        ControlObject object = new ControlObject();
        parser.parse(new File(args[0]), new SAXAutomaton(object));
    }
}

```

УДК 004.738

Р. Э. Асратян, канд. техн. наук, вед. науч. сотр.,
Институт проблем управления им. В. А. Трапезникова РАН, e-mail: rea@ipu.ru

Служба синхронизации процессов в сети

Описываются принципы организации новой Интернет-службы, предназначенной для синхронизации параллельных процессов в глобально-сетевой среде. Описываемая служба может рассматриваться как своего рода "сетевое обобщение" известных механизмов взаимодействия и синхронизации процессов, предоставляемых современными операционными системами (семафоры, критические секции, условные переменные и т. п.) и может быть использована в разработках распределенных информационных систем.

Ключевые слова: распределенные системы, Интернет-службы, синхронизация процессов.

Введение

Средства синхронизации параллельных процессов (семафоры, события, критические секции и т. п.) являются необходимым и привычным для программистов инструментом организации согласованной обработки общих данных несколькими параллельно выполняющимися процессами [1, 2]. Все современные операционные системы (ОС) включают достаточно широкий набор средств синхронизации, действующих в пределах одной машины. В последние годы потребность в подобных средствах стали испытывать и разработчики распределенных систем [3], в которых взаимодействующие процессы могут выполняться на разных сетевых узлах. К сожалению, современные сете-

вые технологии не предлагают каких-либо сетевых аналогов общепринятых средств синхронизации процессов, допускающих применение в многомашиной глобально-сетевой среде.

Рассмотрим, например, задачу "взаимного исключения" процессов, при доступе к общему ресурсу. В одноплатформенных системах эта задача эффективно решается с помощью двоичных семафоров (mutex) или критических секций, обеспечивающих строго последовательный доступ к ресурсу со стороны параллельных процессов. В локально-сетевых, моноплатформенных распределенных системах для этой цели могут быть использованы функции блокировки файлов в разделенных сетевых каталогах (системные вызовы

LockFileEx в MS Windows и flock в UNIX). Отметим, что в обоих случаях обеспечивается квалифицированная форма ожидания доступа к общему ресурсу, при которой ожидающий процесс переводится в неактивное состояние до появления возможности доступа. (Примитивную форму ожидания, основанную на циклическом опросе состояния ресурса с непроизводительным расходом процессорного времени, мы не будем рассматривать в данной статье.)

В разработках глобально-сетевых распределенных систем (т. е. систем, в которых взаимодействующие узлы могут быть удалены друг от друга на сотни и тысячи километров) дело обстоит значительно хуже. Организация разделенного дискового пространства (необходимого для синхронизации с помощью вызовов LockFileEx или flock) в этом случае, как правило, оказывается неприемлемой по соображениям безопасности. (Прямой доступ к разделенным каталогам из глобальной сети практически всегда и совершенно обоснованно рассматривается как фактор риска.) Тем не менее, потребность в синхронизации распределенных по сети процессов существует и в этом случае и может быть связана с согласованным доступом к общим удаленным файлам (например, по протоколу FTP), к разделяемым базам данных или к нересурерабельным вычислительным ресурсам (удаленным исполняемым модулям или процедурам) [4, 5].

В данной статье описывается подход к построению средств синхронизации, которые можно было бы применять, не заботясь о том, на одном ли сетевом узле выполняются взаимодействующие процессы или на разных и размещены ли эти узлы в одной локальной сети или нет. Этот подход основан на создании специальной Интернет-службы — службы синхронизации процессов (ССП), предназначенной для выполнения операций синхронизации независимо от размещения процессов в мультиплатформенной сетевой среде.

Принципы организации службы

Основу ССП составляет сервер ССП — постоянно активная программа, обслуживающая сетевые запросы на выполнение операций синхронизации, исходящие от клиентов. Клиентами ССП являются процессы, заинтересованные во взаимной синхронизации. Эти процессы могут выполняться на том же узле, что и сервер, или на любых других узлах сети. Далее мы будем предполагать, что они выполняются на отдельных рабочих станциях, а сервер ССП — на выделенном компьютере-сервере.

Как и всякий сервер, основанный на TCP/IP [4], сервер ССП постоянно занят "прослушивани-

ем" входящих TCP-соединений на специальном выделенном порте. Зафиксировав входящее соединение, сервер порождает отдельный обслуживающий процесс для обработки входящего запроса, обеспечивает TCP-канал между клиентом и обслуживающим процессом и немедленно возвращается к прослушиванию входящих соединений. (Говоря более строго, для обработки запроса порождается параллельная программная нить — program thread, разделяющая адресное пространство сервера, но в дальнейшем мы будем для краткости использовать термин "процесс".) Вся содержательная обработка запроса (получение входных параметров от клиента по TCP-каналу, собственно обработка запроса и возврат кода завершения клиенту) проводится в обслуживающем процессе. Далее мы рассмотрим принципы сетевой синхронизации в ССП на примере двоичных семафоров, обеспечивающих механизм "взаимного исключения" процессов-клиентов.

Эти принципы проиллюстрированы на рис. 1. В левой части рисунка изображены две рабочие станции, в каждой из которых выполняется по процессу-клиенту (*A* и *B* соответственно), обозначенному белым кружком. В правой части рисунка изображен сервер ССП, в котором организованы несколько двоичных семафоров (mutex), имена которых известны клиентам. (Разумеется, каждый из них предназначен для "регулирования" обращений к какому-то реальному физическому или информационному ресурсу.) Будем считать, что в распоряжении процессов-клиентов имеются специальные клиентские функции "Захватить семафор" и "Освободить семафор", каждая из которых вызывает обращение к серверу ССП по сети с передачей кода операции и имени двоичного семафора (далее — просто семафора) в качестве параметров обращения. Логичность обработки хорошо иллюстрирует следующая последовательность обращений, в которой два процесса-клиента пытаются захватить свободный семафор почти одновременно.

- Процессы *A* и *B* устанавливают соединение с сервером ССП. В результате сервер порождает

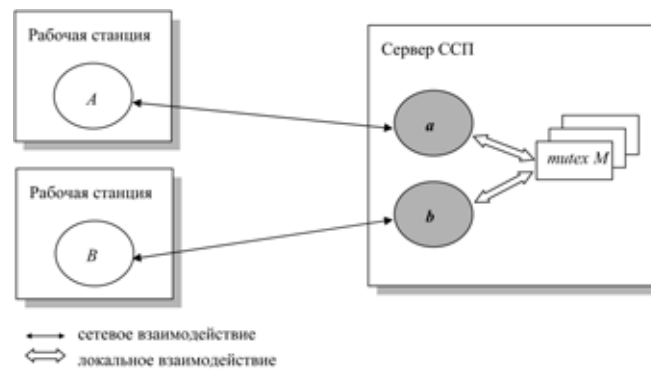


Рис. 1. Обработка запросов к удаленному семафору

два обслуживающих процесса: a и b соответственно, обозначенных серыми кружками.

- Процесс A издает запрос на захват семафора с именем M . Обслуживающий процесс a получает код операции и имя семафора по TSP-каналу, выполняет локальный захват свободного семафора средствами серверной ОС и возвращает код успешного завершения операции процессу A . Во все время обработки процесс A находится в состоянии ожидания чтения кода завершения из TSP-канала. Получив код завершения, процесс A переводится в активное состояние (он становится "владельцем" удаленного семафора).
- Процесс B издает запрос на захват семафора с именем M . Обслуживающий процесс b совершает попытку захватить теперь уже занятый семафор M и переводится в состояние ожидания до его освобождения. В этот момент процесс B продолжает оставаться в состоянии ожидания чтения кода завершения из TSP-канала.
- Процесс A издает запрос на освобождение семафора с именем M . Обслуживающий процесс a выполняет операцию освобождения семафора средствами серверной ОС и возвращает код успешного завершения операции процессу A . В тот же момент процесс b захватывает семафор, переводится в активное состояние и также возвращает код успешного завершения операции процессу B . В результате процесс B переводится в активное состояние (теперь он становится "владельцем" удаленного семафора).
- Процесс B издает запрос на освобождение семафора с именем M . Обслуживающий процесс b выполняет операцию освобождения семафора средствами серверной ОС и возвращает код успешного завершения операции процессу B .

Как видно из приведенного описания, обслуживающий процесс исполняет роль "представителя" процесса-клиента на сервере ССП. Ключевую идею организации службы можно сформулировать так: для процесса-клиента ожидание доступа к удаленному семафору реализуется через ожидания чтения кода завершения из TSP-канала.

Описанная выше организация сервера ССП имеет фундаментальный недостаток — она рассчитана на идеальную сеть и абсолютно непригодна на практике. Причина этого заключается в возможности сетевых сбоев в реальной сети, которые могут привести к разрывам сетевого соединения и непоравимо разрушить всю логику обработки.

Проблема сетевых сбоев

Если после захвата семафора процессом A произойдет разрушение TSP-канала между процессами A и a , то процесс A не сможет издать сетевой

запрос на освобождение семафора. Более того, даже если процесс A восстановит сетевое соединение с сервером ССП, освобождение семафора все равно окажется невозможным. Ведь в этом случае будет порожден новый обслуживающий процесс a' , который не сможет выполнить операцию освобождения, так как с точки зрения серверной ОС он не является владельцем семафора. Подобная же проблема возникает в случае разрушения TSP-канала между процессами B и b в период, когда процесс B ожидает доступ к семафору. После освобождения семафора серверная ОС предоставит его процессу b , но связь между процессами B и b к этому времени уже разрушена и невозможна. (Кстати, именно по этой причине невозможно построить аналогичный механизм семафоров на основе функций блокировки объектов, имеющихся в современных СУБД, так как все такие блокировки автоматически снимаются при завершении или разрыве сетевого соединения.)

Причина указанных выше проблем коренится в общей "слабости" практически всех Интернет-служб, работающих в режиме on-line: "контекст" сеанса связи, как правило, жестко ассоциируется с сетевым соединением. При разрушении соединения теряется контекст и, как следствие, результаты обработки (наверное, многим знакома ситуация, когда вследствие разрыва сетевого соединения с Web-сервером результаты длительной работы, например CGI-программы, оказываются безвозвратно потерянными). В службе синхронизации процессов эта "слабость" приводит не только к потере результатов, но к нарушению логики обработки.

Можно сформулировать два вывода из приведенного обсуждения.

- Служба ССП нуждается в понятии устойчивого "контекста" взаимодействия, не разрушающегося при разрывах (случайных или намеренных) сетевого соединения клиента с сервером. Этот "контекст" может быть введен в форме уникального (в пределах сервера) идентификатора сеанса связи, который процесс-клиент получает от сервера в результате выполнения открытия сеанса связи и который он предъявляет во всех последующих запросах к серверу вплоть до закрытия сеанса.
- Учет занятости семафора на сервере ССП должен проводиться не в терминах идентификаторов обслуживающих процессов в серверной ОС, а в терминах идентификаторов сеансов. Другими словами, обслуживающий процесс a' должен иметь право освободить семафор, занятый процессом a , если оба процесса запущены в рамках одного сеанса связи (т. е. фактически семафор должен захватываться не процессом, а сеансом). Это означает, что сервер ССП не мо-

жет пользоваться готовым механизмом двоичных семафоров, предоставляемым серверной ОС, а должен поддерживать свой собственный механизм.

Организация с устойчивым контекстом

Дополним описанную выше организацию сервера ССП новой функциональностью, связанной с обработкой сеансов связи. Дополненная организация базируется на следующих принципах.

- В распоряжение процесса-клиента предоставляются два новых запроса к серверу ССП: "Открыть сеанс связи" и "Закрыть сеанс связи". В результате выполнения первого из них процесс-клиент получает уникальный идентификатор открытого сеанса связи. Запросы на захват или освобождение семафора могут издаваться только после открытия сеанса, причем идентификатор сеанса включается в параметры этих запросов наряду с именем семафора. Таким образом, каждый запрос связывается с определенным сеансом.
- Сервер ССП ведет учет всех открытых сеансов и процессов таким образом, что каждый обслуживающий процесс ассоциируется с переданным ему (через параметры запроса) идентификатором сеанса. Каждый занятый семафор также ассоциирован с идентификатором сеанса захватившего его процесса и с очередью обслуживающих процессов, дожидаящихся доступа к этому семафору. (Разумеется, все это реализуется с помощью довольно сложной структуры данных сервера ССП, включающей связанные между собой дескрипторы сеансов, обслуживающих процессов и семафоров, но мы не будем останавливаться на описании этой структуры.)
- После завершения обработки каждого запроса обслуживающий процесс помещает код завершения в дескриптор сеанса связи и, кроме того, передает его клиенту по ТСР-каналу. Если обслуживающий процесс обнаруживает, что соединение с процессом-клиентом разорвано (например, при попытке передачи кода завершения), он немедленно заканчивает работу. Завершение обслуживающего процесса или разрыв соединения никогда не влекут за собой автоматическое освобождение семафора.
- Если при издании запроса процесс-клиент обнаруживает, что соединение с сервером ССП разрушено, он автоматически восстанавливает его (с порождением нового процесса на сервере). Если разрыв соединения произошел в период ожидания кода завершения от обслуживающего процесса, процесс-клиент немедленно издает новый специальный запрос к серверу

ССП: "Продолжить ожидание окончания обработки" и возвращается в состояние ожидания кода завершения. (Это поведение заложено в клиентские функции отправки запросов и реализуется автоматически.)

- Запрос "Продолжить ожидание окончания обработки" порождает новый процесс на сервере ССП — процесс ожидания. Используя идентификатор текущего сеанса связи, этот процесс отыскивает обслуживающий процесс, потерявший соединение с процессом-клиентом. Если этот процесс уже завершен, то процесс ожидания попросту считывает его код завершения из дескриптора сеанса связи, передает этот код процессу-клиенту по ТСР-каналу и заканчивает свою работу. В противном случае он переходит в состояние ожидания завершения обслуживающего процесса. Эта операция может итеративно повторяться несколько раз: если процесс ожидания также потеряет связь с процессом-клиентом, то будет порожден новый процесс ожидания и т. д. На рис. 2 изображена ситуация, когда в рамках сеанса связи порождено несколько процессов (обозначены серыми кружками), один из которых является обслуживающим, а остальные — процессами ожидания.
- После завершения обработки обслуживающий процесс проверяет, имеются ли процессы ожидания, связанные с тем же сеансом связи. Если есть, он посылает сигнал активизации всем этим процессам и заканчивает свою работу. Тот из процессов ожидания, который имеет связь с процессом-клиентом, передает ему код завершения обслуживающего процесса по ТСР-каналу, после чего все процессы ожидания, порожденные в рамках данного сеанса, также заканчивают свою работу.
- При захвате семафора он связывается с сеансом, к которому относится соответствующий запрос. При выполнении запроса на освобождение семафора проводится проверка, отно-

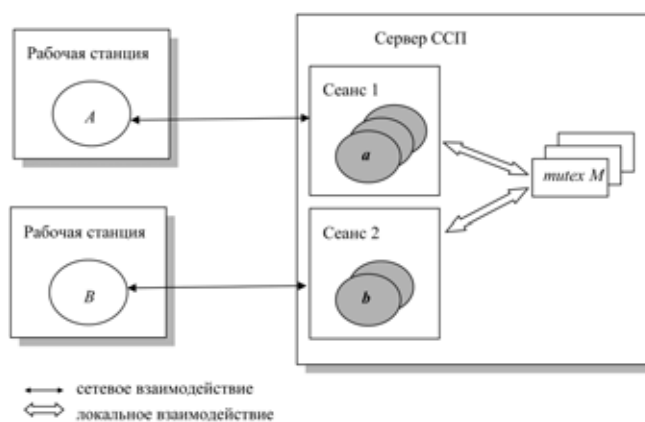


Рис. 2. Обработка сеансов связи

сится ли этот запрос к тому же сеансу, с которым связан семафор. Если не относится, то пользователю возвращается код ошибки. При выполнении запроса на захват занятого семафора также проводится проверка, относится ли этот запрос к тому же сеансу, с которым связан семафор. Если относится, то пользователю также возвращается код ошибки.

Управление состоянием и взаимодействием обслуживающих процессов и процессов ожидания на сервере ССП осуществляется с помощью локальных средств синхронизации процессов, предоставляемых серверной ОС (например, с помощью механизма событий в Win32 или условных переменных для синхронизации программных нитей в UNIX).

Поведение клиента

Для взаимодействия с сервером ССП программа клиента должна использовать специальную клиентскую библиотеку функций, в которой определен класс SSPClient, содержащий в себе все необходимые для этого методы (функции-члены). Чтобы проиллюстрировать работу клиента с удаленным семафором, рассмотрим следующий "скелет" кода в нотации языка C++.

```
char SessionId [20]; // Строка для хранения идентификатора сеанса
SSPClient Cssp; // Объект класса SSPClient
Cssp.Config("192.168.1.77", 8132, "alibaba", "sezam");
// Определяем // конфигурацию
Cssp.Connect(); // Соединение с сервером ССП
Cssp.OpenSession(SessionId); // Открытие сеанса
Cssp.GetMutex(SessionId, "main_mutex"); // Захват семафора
...
// Обработка разделяемого ресурса в монопольном режиме
...
Cssp.ReleaseMutex(SessionId, "main_mutex"); // Освобождение семафора
Cssp.CloseSession(SessionId); // Закрытие сеанса
Cssp.Disconnect(); // Закрытие соединения с сервером ССП
```

Первые две строки определяют две переменные: строку символов SessionId и объект класса SSPClient. Третья строка настраивает объект на работу с определенным сервером ССП. В качестве параметров объекту передаются IP-адрес (или Интернет-имя) сервера, номер порта, имя и пароль пользователя (сервер ССП обслуживает только зарегистрированных пользователей).

Четвертая строка устанавливает TCP-соединение с сервером, а пятая — открывает на сервере новый сеанс связи: в результате выполнения метода OpenSession строка SessionId окажется заполненной уникальным (для конкретного сервера) идентификатором открытого сеанса. Последующие методы (GetMutex — захват семафора и ReleaseMutex — освобождение семафора) будут ис-

пользовать этот идентификатор в качестве параметра. Если при вызове метода GetMutex семафор окажется занят другим процессом, выполняемым на том же или на удаленном узле, клиентская программа перейдет в состояние ожидания до момента освобождения семафора. Физическая природа разделяемого ресурса и методы доступа к нему могут быть произвольными и не затрагивают механизма синхронизации. Пример заканчивается вызовами методов освобождения семафора, закрытия сеанса и закрытия соединения с сервером. Разумеется, из данного фрагмента кода намеренно удалены операторы проверки успеха выполнения методов и обработки исключений.

Отметим, что логика работы клиента ССП с семафорами не разрушается в результате разрыва сетевого соединения. Так, например, клиент мог бы намеренно закрыть соединение с сервером после выполнения операции захвата семафора (метод GetMutex) с помощью метода Disconnect. В этом случае сетевое соединение и возможность продолжения обработки были бы автоматически восстановлены при выполнении метода ReleaseMutex (с тем же значением параметра SessionId).

Вместе с тем, необходимо подчеркнуть особую важность корректного поведения клиента в ССП. Если клиент по какой-либо причине своевременно не освободит захваченный семафор или не закроет сеанс связи, то это может привести к блокировке доступа к ресурсу других клиентов или к напрасному расходованию ресурсов сервера. Даже введение ограничений (тайм-аутов) на время владения ресурсом и время существования сеанса вместе с процедурой "сборки мусора" для автоматического закрытия слишком давно открытых сеансов на сервере ССП лишь частично решает проблему. Поэтому клиенты ССП должны программироваться особенно тщательно с обязательным принятием мер для освобождения семафора в случае аварийного завершения. Например, приведенный ниже фрагмент кода в нотации C++ иллюстрирует применение конструкций try и catch для обработки аварийной ситуации в программе клиента. (Предполагается, что вся работа с семафорами и разделяемыми ресурсами сосредоточена в блоке try, а любая исключительная ситуация приводит к автоматической передаче управления в блок catch.):

```
try
{
    // Работа с семафорами и ресурсами
}
catch (...)
{
    Cssp.ReleaseAllMutexes(SessionId); // Освобождение всех семафоров
    Cssp.CloseSession(SessionId); // Закрытие сеанса
}
```

Использованный в блоке catch метод Release-AllMutexes специально предназначен для освобождения всех семафоров, захваченных (и еще не освобожденных) в течение сеанса на сервере ССП. (Как и в предыдущем примере, предполагается, что идентификатор сеанса связи сохраняется в строковой переменной SessionId, а Cssp — объект класса SSPClient).

Заключение

К недостаткам ССП можно отнести уже отмеченную "чувствительность" к некорректному поведению клиентов (в какой-то мере она является "платой" за устойчивость к разрывам соединения). Во всяком случае, ССП предъявляет более высокие требования к корректности и отлаженности клиентских программ, чем большинство Интернет-служб, так как неверное поведение одного клиента может исказить работу остальных.

Описанный метод синхронизации процессов может быть полезен при разработке территориально-распределенных систем и приложений с гибким распределением обрабатываемых процессов по сетевым узлам. В этом случае особенно важны универсальные механизмы синхронизации, не зависящие от того, на одном узле выполняются синхронизируемые процессы или на географически удаленных узлах.

За рамками данной статьи остались такие вопросы как защита ССП от несанкционированного доступа, администрирование сервера ССП или

конкретные форматы данных в TCP-канале, которые, конечно, очень важны для практики, но слабо касаются сути рассматриваемой проблемы.

Экспериментальная реализация сервера ССП была проведена на двух платформах: в форме сервиса для Windows Server 2000/2003 [6] и в форме "демона" для UNIX FreeBSD [7]. Описанный механизм двоичных семафоров усилен возможностью выбора монопольного или разделенного режимов работы с ресурсом, а также указания предельного времени ожидания (тайм-аута). Кроме того, реализация сервера ССП поддерживает механизм очередей сообщений, обеспечивающий обмен данными между процессами на разных сетевых узлах с необходимой синхронизацией процессов-получателей с сообщениями, построенной по аналогичным принципам.

Список литературы

1. Дijkstra Э. Взаимодействие последовательных процессов // Языки программирования. М.: Мир, 1972. 406 с.
2. Таненбаум Э. Современные операционные системы. СПб.: Питер, 2002. 1040 с.
3. Гофф М. Сетевые распределенные вычисления: достижения и проблемы. М.: КУДИЦ-ОБРАЗ, 2005. 320 с.
4. Джамса К., Коуп К. Программирование для Интернет в среде Windows. СПб.: Питер, 1996. 659 с.
5. Асратян Р. Э. Интернет-служба для поддержки распределенных вычислений // Информационные технологии. 2006. № 12. С. 60—66.
6. Андреев А. Г., Беззубов Е. Ю., Емельянов М. М., Кокорева О. И., Чекмарев А. Н. Windows 2000: Server и Professional. СПб.: БХВ—Санкт-Петербург, 2001. 1055 с.
7. Келли-Бутл С. Введение в Unix. М.: ЛОРИ, 1995. 596 с.

МОДЕЛИРОВАНИЕ

УДК 004.2

Оцоков Ш. А., канд. техн. наук, доц.,

Московский энергетический институт (технический университет), e-mail: shamil24@mail.ru

Применение модулярной арифметики с фиксированной точкой для ослабления влияния ошибок округления компьютерных вычислений

Предложена модулярная арифметика с фиксированной точкой для исключения катастрофической потери точности при вычислениях с разномасштабными величинами. Эффект катастрофической потери точности продемонстрирован на примере определения скалярного произведения двух векторов и дискретного преобразования Фурье.

Ключевые слова: модулярная арифметика, модулярный формат, рабочий положительный диапазон, рабочий отрицательный диапазон.

Введение

Арифметика с плавающей точкой, в которой проводится большинство компьютерных вычислений, обладает рядом недостатков. Один из них — неравномерное распределение чисел внутри диапазона (рис. 1).

На рис. 1 видно, что плотность распределения чисел с плавающей точкой неравномерная — велика вблизи нуля и уменьшается по мере приближения к границам диапазона $[-F, F]$. Вследствие этого может происходить катастрофическая потеря точности компьютерных вычислений при:

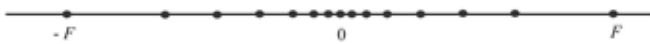


Рис. 1. Неравномерное распределение чисел с плавающей точкой

Максимальная относительная погрешность ДПФ

n = 5		n = 100	
α	$\delta, \%$	α	$\delta, \%$
10^2	$7 \cdot 10^{-12}$	10^2	$1 \cdot 10^{-12}$
10^9	$2 \cdot 10^{-5}$	10^9	$1 \cdot 10^{-5}$
10^{11}	$1 \cdot 10^{-2}$	10^{11}	$4 \cdot 10^{-3}$
10^{14}	1	10^{14}	32

- определении скалярного произведения двух векторов со значительно различающимися координатами;
- дискретном преобразовании Фурье;
- численном решении систем дифференциальных уравнений в частных производных с сильно различающимися коэффициентами (моделировании физических процессов в полупроводниках, в нанoeлектронике и др. [2]) и др.

Рассмотрим некоторые примеры катастрофической потери точности в арифметике с плавающей точкой.

1. Пусть даны векторы x, y с координатами

$$x = (10^{17}, 1223, 10^{18}, 10^{15}, 3, -10^{12});$$

$$y = (10^{20}, 2, -10^{19}, 10^{13}, 2111, 10^{16}).$$

Скалярное произведение этих векторов, вычисленное в арифметике с плавающей точкой одинарной точности, равно $4.61168602259351E + 0020$, а истинное значение 8779.

2. Известно, что дискретное преобразование Фурье (ДПФ) вектора x в вектор y осуществляется по формуле [4]

$$y_k = \frac{1}{n} \sum_{j=0}^{n-1} x_j \left[\cos \frac{2\pi jk}{n} - i \sin \frac{2\pi jk}{n} \right],$$

$$0 \leq k \leq n - 1,$$

обратное дискретное преобразование Фурье — по формуле

$$x_k = \sum_{j=0}^{n-1} y_j \left[\cos \frac{2\pi jk}{n} + i \sin \frac{2\pi jk}{n} \right],$$

$$0 \leq k \leq n - 1.$$

Пусть задан вектор $x = (x_1, x_2, x_3, \dots, x_n)$ с координатами, определяемыми следующим образом:

$$x_i = \begin{cases} i/10, & \text{если } i \text{ — нечетное;} \\ i/10 + \alpha, & \text{если } i \text{ — четное,} \\ i = 1, \dots, n, \alpha > 0. \end{cases}$$

Если провести прямое, затем обратное дискретное преобразование Фурье вектора $x = (x_1, x_2, x_3, \dots, x_n)$, получим вектор $x^* = (x_1^*, x_2^*, x_3^*, \dots, x_n^*)$, который должен приблизительно совпадать с x . Оценить максимальную относительную погрешность преобразований можно по формуле

$$\delta = \max \left(\frac{|x_1^* - x_1|}{x_1} 100, \frac{|x_2^* - x_2|}{x_2} 100, \dots, \frac{|x_n^* - x_n|}{x_n} 100 \right).$$

В таблице приведены результаты таких оценок для различных α, n .

Из таблицы видно, что с увеличением параметра α возрастает максимальная относительная погрешность ДПФ, так, например, для 100-точечного ДПФ при $\alpha = 10^{14}$ она составляет 32 %.

Цель настоящей статьи состоит в разработке алгоритмов выполнения арифметических операций, которые *гарантированно исключают* катастрофическую потерю точности при компьютерных вычислениях с разномасштабными величинами. Рассмотрим предлагаемые алгоритмы выполнения арифметических операций.

Модулярная арифметика с фиксированной точкой

Известно, что распределение чисел формата с фиксированной точкой внутри диапазона $[-F, F]$ равномерное (рис. 2).

Недостатком формата с фиксированной точкой по сравнению с традиционным является необходимость хранения и обработки чисел большой разрядности. Так, например, для представления чисел с плавающей точкой в формате с фиксированной точкой двойной точности ($[-1,7E308 \dots 1/7E308]$) требуется 308 десятичных разрядов и, как следствие этого, — низкое быстродействие выполнения арифметических операций.

Вычисления с числами формата с фиксированной точкой могут быть ускорены, если они будут выполняться в модулярной системе счисления (МСС). В модулярной системе счисления любые числа представляются в виде набора остатков от деления на заранее выбранные простые числа (модули), и арифметические операции выполняются не с исходными числами, а с их остатками. Достоинством и главным преимуществом модулярной системы счисления является отсутствие переносов между модулями, что позволяет распараллелить процесс выполнения арифметических операций по всем модулям, в позиционной системе счисления могут быть межразрядные пере-

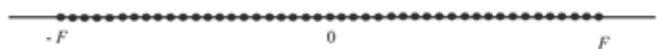


Рис. 2. Равномерное распределение чисел с фиксированной точкой

носы. Арифметические операции в модулярной системе счисления могут выполняться параллельно и независимо по каждому модулю, благодаря чему достигается сильное ускорение вычислений с многоразрядными числами.

Суть предлагаемого способа состоит в обобщении алгоритмов модулярной арифметики: сложения, вычитания, умножения и деления на константу для работы с числами, представленными в формате с фиксированной точкой.

Рассмотрим **модулярный формат** представления двоичных дробей в МСС [3]. Любое число A в формате с фиксированной точкой можно представить в виде

$$\text{или} \quad A = \frac{\underbrace{xx \dots x}_{n_f} \cdot \underbrace{y \dots y}_{k_f}}{2^{k_f}}, \quad (1)$$

где K — целое число такое, что $|K| \leq 2^{n_f+k_f} - 1$.

Пусть P_1, P_2, \dots, P_n — модули (последовательность простых чисел) таких, что $P_1 < P_2 < \dots < P_n$, $P_1 > 2$ и $P = P_1 \cdot P_2 \cdot \dots \cdot P_n$ — их произведение. Пусть A — число вида

$$A = \frac{K}{2^t},$$

где $0 \leq t \leq k_f$; K не кратно двум.

Будем представлять число A в модулярной системе счисления (МСС) в виде

$$A = [(\alpha_1, \alpha_2, \dots, \alpha_n), t],$$

где

$$\alpha_i \equiv |A|_{P_i} \equiv |K \cdot 2^{-t}|_{P_i}, \quad i = 1, \dots, n - \text{мантисса}; \quad (2)$$

t — порядок числа; $|x|_p$ — остаток от деления числа x на p .

Справедлива следующая теорема.

Теорема. Пусть даны числа $A_1 = \frac{K_1}{2^{t_1}}, A_2 = \frac{K_2}{2^{t_2}}$,

$C_1 = A_1 \pm A_2, C_2 = A_1 \cdot A_2$, и их модулярное представление имеет вид

$$\begin{aligned} A_1 &= [(\alpha_1, \alpha_2, \dots, \alpha_n), t_1]; \\ A_2 &= [(\beta_1, \beta_2, \dots, \beta_n), t_2]; \\ C_1 &= [(\gamma_1, \gamma_2, \dots, \gamma_n), t_3]; \\ C_2 &= [(\eta_1, \eta_2, \dots, \eta_n), t_4], \end{aligned}$$

где

$$\alpha_i \equiv |K_1 \cdot 2^{-t_1}|_{P_i}, \quad \beta_i \equiv |K_2 \cdot 2^{-t_2}|_{P_i}, \quad i = 1, \dots, n.$$

Тогда

$$\begin{aligned} \gamma_i &\equiv |\alpha_i \pm \beta_i|_{P_i}; \\ t_3 &= \max(t_1, t_2); \\ \eta_i &\equiv |\alpha_i \cdot \beta_i|_{P_i}; \\ t_4 &= t_1 + t_2. \end{aligned}$$

Доказательство. Из условия теоремы

$$\begin{aligned} C_1 &= A_1 + A_2 = \\ &= \frac{K_1}{2^{t_1}} + \frac{K_2}{2^{t_2}} = \frac{K_1 \cdot 2^{\max(t_1, t_2) - t_1} + K_2 \cdot 2^{\max(t_1, t_2) - t_2}}{2^{\max(t_1, t_2)}}. \end{aligned}$$

По определению числа γ_i (2)

$$\begin{aligned} \gamma_i &\equiv |C_1|_{P_i} \equiv \left| \frac{K_1 \cdot 2^{\max(t_1, t_2) - t_1} + K_2 \cdot 2^{\max(t_1, t_2) - t_2}}{2^{\max(t_1, t_2)}} \right|_{P_i} \equiv \\ &\equiv \left| |K_1 \cdot 2^{-t_1}|_{P_i} + |K_2 \cdot 2^{-t_2}|_{P_i} \right|_{P_i} \equiv |\alpha_i \pm \beta_i|_{P_i}, \\ & \quad i = 1, \dots, n. \end{aligned}$$

и $t_3 = \max(t_1, t_2)$.

Аналогично доказывается вычитание чисел A_1, A_2 .

Пусть

$$C_2 = A_1 \cdot A_2 = \frac{K_1}{2^{t_1}} \cdot \frac{K_2}{2^{t_2}} = \frac{K_1 \cdot K_2}{2^{t_1 + t_2}},$$

тогда

$$\begin{aligned} \eta_i &\equiv |C_2|_{P_i} \equiv \left| \frac{K_1 \cdot K_2}{2^{t_1 + t_2}} \right|_{P_i} \equiv \\ &\equiv \left| |K_1 \cdot 2^{-t_1}|_{P_i} \cdot |K_2 \cdot 2^{-t_2}|_{P_i} \right|_{P_i} \equiv |\alpha_i \cdot \beta_i|_{P_i}, \\ & \quad i = 1, \dots, n, \end{aligned}$$

и $t_4 = t_1 + t_2$,

что и требовалось доказать.

Правило определения дроби, соответствующей заданному числу в модулярной системе счисления

Для представления как положительных, так и отрицательных чисел разобьем интервал $[0, \dots, P)$ на четыре части (рис. 3):

$$\begin{aligned} &[0, \dots, P_{(1/4)}), [P_{(1/4)}, \dots, P_{(1/2)}), \\ &[P_{(1/2)}, \dots, P'_{(1/4)}), [P'_{(1/4)}, \dots, P), \end{aligned}$$

$$\begin{aligned} \text{где} \quad P &= \prod_{i=1}^N P_i, \quad P_{(1/2)} = \prod_{i=1}^{N-1} P_i, \\ P_{(1/4)} &= \prod_{i=1}^{\lfloor (N-1)/2 \rfloor} P_i; \quad (3) \end{aligned}$$



Рис. 3. Разбиение отрезка

$$P'_{(1/4)} = P - P_{(1/4)};$$

$\lfloor x \rfloor$ — целая часть x ; N — число модулей.

Выберем модули МСС таким образом, чтобы выполнялось неравенство

$$P_{(1/4)} > 2^{(n_f + k_f)}. \quad (4)$$

Из (3) и условия $P_1 < P_2 < \dots < P_n$ следует, что $P_{(1/4)}^2 < P_{(1/2)}$.

Пусть $A_1 = [(\alpha_1, \alpha_2, \dots, \alpha_n), t]$ — число, соответствующее искомой дроби, представленной в формате с фиксированной точкой с длинами целой и дробной частей соответственно n_f, k_f . Тогда числитель искомой дроби K однозначно определяется исходя из следующей системы сравнений (так как из (4), следует, что $K < P$):

$$\begin{cases} |K|_{P_1} \equiv |\alpha_1 \cdot 2^t|_{P_1} \equiv \lambda_1; \\ |K|_{P_2} \equiv |\alpha_2 \cdot 2^t|_{P_2} \equiv \lambda_2; \\ \dots \\ |K|_{P_n} \equiv |\alpha_n \cdot 2^t|_{P_n} \equiv \lambda_n. \end{cases} \quad (5)$$

Интервал $[0, \dots, P_{(1/4)})$ определяет диапазон, в котором представляются все положительные числа в МСС, назовем его **рабочим положительным диапазоном** (РПД). В МСС отрицательное число представляется в виде дополнения до модуля, т. е. $|-K|_P = |P - K|_P$. Поэтому **рабочий отрицательный диапазон** (РОД), в котором представляются все отрицательные числа, — это интервал $[P'_{(1/4)}, \dots, P)$.

Если число K принадлежит РПД, то K положительное, если K принадлежит РОД, то K отрицательное, и если K не принадлежит ни РОД, ни РПД, то эта ситуация сигнализирует об ошибке **не-реполнения** (выхода за пределы допустимых диапазонов для положительных и отрицательных чисел).

Для того чтобы определить принадлежность числа K РПД или РОД, его необходимо сравнить с $P_{(1/4)}$. Способ сравнения чисел в МСС, который используется в настоящей работе, основан на представлении чисел в смешанной системе счисления: сравниваемые числа переводятся в смешанную систему счисления, в которой они легко сравниваются.

Пусть

$$K = (\lambda_1, \lambda_2, \dots, \lambda_n),$$

где λ_i определяются из системы сравнений (5).

Преобразуем K в смешанную систему счисления на основе алгоритма, описанного в [1]. Получим представление K в смешанной системе счисления в виде

$$K = \sum_{i=0}^{n-1} v_i \cdot a_i, \quad (5)$$

где $v_i = \prod_{j=0}^i P_j$, $0 < i < n-1$ — основания смешанной системы счисления [1]; a_i — цифры разложения K в системе счисления со смешанными основаниями, P_i — модули ($i = 0, \dots, n-1$) и $P_0 = 1$.

Если $K < P_{(1/4)}$, то $A > 0$ и K принадлежит рабочему положительному диапазону. Числа $P_{(1/2)}$, $P_{(1/4)}$ в системе счисления со смешанными основаниями представляются в виде

$$(0, 0, \dots, 1, \dots, 0),$$

где цифра "1" будет на позициях $N-2$ и $\lfloor (N-1)/2 \rfloor - 1$ соответственно, начиная счет от нуля. Поэтому после представления K в смешанной системе счисления его легко можно сравнить с $P_{(1/4)}$.

Если $K > P'_{(1/4)}$, то $A < 0$ и K принадлежит рабочему отрицательному диапазону. Определить принадлежность КРОД можно сравнением чисел K , $P'_{(1/4)}$ или K^* , $P_{(1/4)}$, где

$$K^* = (P_1 - \lambda_1, P_2 - \lambda_2, \dots, P_n - \lambda_n).$$

Если $K^* < P_{(1/4)}$, то $A < 0$ принадлежит РОД и его округление не требуется. Преобразуя K^* в смешанную систему счисления, его легко можно сравнить с $P_{(1/4)}$.

Знаменатель искомой дроби есть число 2^t .

Рассмотрим случай, когда искомая дробь является сократимой, т. е.

$$A = \frac{K}{2^t} = \frac{K_1}{2^{t_1}},$$

где

$$t_1 > t, K_1 = K \cdot 2^{t_1 - t}.$$

Тогда A в МСС соответствует число

$$A_1 = [(\alpha_1, \alpha_2, \dots, \alpha_n), t_1] = [(\alpha_1, \alpha_2, \dots, \alpha_n), t],$$

так как

$$|K_1|_{P_1} \equiv |\alpha_1 \cdot 2^{t_1}|_{P_1} \equiv \lambda_1;$$

$$|K_1|_{P_2} \equiv |\alpha_2 \cdot 2^{t_1}|_{P_2} \equiv \lambda_2;$$

...

$$|K_1|_{P_n} \equiv |\alpha_n \cdot 2^{t_1}|_{P_n} \equiv \lambda_n;$$

а числа удовлетворяют сравнениям

$$\begin{aligned}\alpha_i &\equiv |A|_{P_i} \equiv |K \cdot 2^{-t}|_{P_i} \equiv \\ &\equiv |K \cdot 2^{t_1-1} \cdot 2^{-t_1}|_{P_i} \equiv |K_1 \cdot 2^{-t_1}|_{P_i}.\end{aligned}$$

Алгоритмы модулярной арифметики с фиксированной точкой были программно реализованы и использовались для вычисления скалярного произведения векторов

$$\begin{aligned}x &= (10^{17}, 1223, 10^{18}, 10^{15}, 3, -10^{12}), \\ y &= (10^{20}, 2, -10^{19}, 10^{13}, 2111, 10^{16})\end{aligned}$$

при следующих параметрах:
выбранные модули

(9999889, 9999901, 9999907, 9999929, 9999931,
9999937, 9999943, 9999971, 9999973, 9999991,
99999827, 99999839, 99999847, 99999931,
99999941, 99999959);

$n_f = 37$; $k_f = 10$ (длина целой и дробной частей числа в формате с фиксированной точкой была выбрана так, чтобы представить максимальный возможный промежуточный результат — число 10^{37}).

Скалярное произведение, вычисленное с использованием предложенных алгоритмов, было равно числу 8779, что совпадает с истинным значением.

При реализации дискретного преобразования Фурье значения тригонометрических функций синус и косинус представлялись в виде десятичных

дробей с точностью 20 знаков после запятой. Применение алгоритмов модулярной арифметики с фиксированной точкой для дискретного преобразования Фурье позволило исключить катастрофическую потерю точности результатов для всех α , n .

Заключение

Вычисления скалярного произведения двух векторов и дискретного преобразования Фурье показывают, что расчеты, выполняемые в традиционной компьютерной арифметике, вследствие неравномерного распределения чисел с плавающей точкой могут приводить к катастрофической потере точности результатов. В настоящей работе предложен универсальный подход исключения катастрофической потери точности компьютерных вычислений с разномасштабными величинами, основанный на применении модулярной арифметики с фиксированной точкой.

Список литературы

1. **Грегори Р., Кришнамурти Е.** Безошибочные вычисления. Методы и приложения. М.: Мир, 1988. 207 с.
2. **Филатов Н. И., Чарыков Н. А.** Моделирование на ЭВМ полупроводниковых диодов: Учеб. пособие по курсу "Физика полупроводниковых приборов". М.: МЭИ, 1985. 52 с.
3. **Оцоков Ш. А.** Применение модулярной арифметики для решения непрерывной задачи наименьших квадратов. // Труды 3-й Международной научно-технической конференции "Инфоком-3", Ставрополь 2008.
4. **Дискретное преобразование Фурье.** [http:// ru.wikipedia.org/wiki/ Дискретное_преобразование_Фурье](http://ru.wikipedia.org/wiki/Дискретное_преобразование_Фурье)

УДК 621.39

А. Н. Голубинский, канд. техн. наук, ст. преподаватель,
Воронежский институт МВД России, e-mail: annikgol@mail.ru

Методика проверки на адекватность математической модели речевого сигнала экспериментальным данным

Приведены и проанализированы большинство используемых мер соответствия математической модели речевого сигнала экспериментальным данным. Разработана методика проверки математической модели речевого сигнала на адекватность при использовании строгого статистического критерия.

Ключевые слова: математическая модель, адекватность, речевой сигнал, мера соответствия, ошибка модели, статистический критерий.

На сегодняшний день кажется немыслимым существование цивилизованного человека без технологий передачи, обработки и хранения информации. При этом речевые сигналы как важные носители сообщений являются неотъемлемой частью информационных технологий, например, в задачах кодирования, сжатия, передачи речи, а

также идентификации и верификации диктора по голосу [1].

Для решения задач, связанных с обработкой и передачей речевой информации, возникает необходимость описания речевых сигналов. Это, как правило, проводится двумя способами, первый из которых основывается на описании речевых сиг-

налов с помощью регистрации его характеристик, где применяется хранение (обработка и сравнение) большого массива данных [2—4]. Другой способ описания речевых сигналов связан с построением различных математических моделей речевых сигналов, адекватно описывающих речевой сигнал с требуемой точностью [2, 5, 6].

Однако, к сожалению, в настоящее время отсутствует литература, в которой были бы приведены и систематизированы меры соответствия модели экспериментальным данным применительно к построению математических моделей речевых сигналов.

Таким образом, представляет научный интерес разработка методики проверки математической модели речевого сигнала на адекватность.

Цель работы — обзор и систематизация используемых мер соответствия математической модели речевого сигнала экспериментальным данным, а также разработка методики проверки математической модели речевого сигнала на адекватность при использовании объективно-статистического критерия проверки гипотез.

Анализ математических моделей речевых сигналов в виде функциональной зависимости от времени

Решение задачи о системном изложении мер соответствия (ошибок модели или невязок) математической модели речевого сигнала экспериментальным данным является довольно обширной темой для научного исследования. Это связано прежде всего с возможным многообразием описаний речевых сигналов. Приведем четыре основных подхода к созданию математических моделей речевых сигналов, заданных функциональной зависимостью отсеченных значений модели от времени.

1. Линейное предсказание речевого сигнала. Недостатками данного подхода являются малое время предсказания, а также тот факт, что получение отсеченных временных значений сигнала математической модели основано на использовании дополнительного источника сигнала в виде белого шума, что влечет за собой определенные трудности.

2. Решение дифференциальных уравнений (модифицированное волновое уравнение и граничные условия), составленных при использовании нескольких условных физических приближений. Эти уравнения описывают распространение акустического колебания в речевом тракте с учетом возбуждающего сигнала. В данном подходе присутствуют следующие основные недостатки: при аппроксимации возбуждающего сигнала и площади поперечного сечения речевого аппарата, если и удастся, как правило, численными методами решить данное волновое уравнение, то полученные результаты имеют очень сложный характер пространственно-временной зависимости и

малопригодны для быстрого и адекватного анализа речи.

3. Гармоническая математическая модель речевого сигнала, использующая для представления сигнала синусоиды кратной частоты (оставшаяся часть сигнала, которая не может быть описана с помощью данной модели, называется шумовой). Представление сигнала в форме гармоники плюс шум эффективно применяется во многих речевых приложениях [7, 8]. Следует отметить, что от разделения математической модели речевого сигнала на периодическую (вокализованные участки речи) и шумовую (невокализованная речь) части в существенной степени зависит адекватность модели и ее применимость в конкретных технических задачах [2]. Данную модель можно синтезировать, используя, например, коэффициенты дискретного преобразования Фурье речевого сигнала. Основные недостатки: использование, как правило, большого числа коэффициентов модели, а также ограниченная точность модели, связанная с бесконечной длительностью гармонической функции.

4. Построение математической модели речевого сигнала в виде явной функциональной временной зависимости, основанное на теории модуляции (как частный случай — модулированный импульс [9]).

В данной работе рассматриваются меры соответствия (ошибки) модели применительно к математическим моделям речевых сигналов, заданных в явном виде функциональной зависимостью от времени. В качестве примера для анализа ошибок модели будем использовать быстро развивающийся подход построения математических моделей, основанный на теории модуляции. Данный подход быстро становится популярным ввиду высокой точности описания речевых сигналов [9, 10], ясной физической интерпретации математической модели на основе физических принципов речеобразования [11, 12], а также относительно простой реализации алгоритмов обработки [13, 14].

Основные этапы моделирования

Говоря о применимости моделей к описанию реальной действительности, мы подразумеваем возможность их практического использования. Приведем основные этапы моделирования [15].

На первом (исходном) этапе должны быть определены: конечные цели моделирования; набор факторов и показателей, взаимосвязи между которыми нас интересуют; роли этих факторов и показателей — какие из них в рамках поставленной конкретной задачи можно считать входными, а какие — выходными.

На втором этапе приступают к постулированию, математической формализации, и, если возможно, к экспериментальной проверке ряда есте-

ственных исходных допущений, относящихся к природе и качественному характеру "физики" исследуемого явления.

Третий этап может быть назван собственно моделирующим, так как он включает в себя непосредственный вывод общего вида модельных соотношений, связывающих между собой интересующие нас входные и выходные показатели.

Четвертый этап моделирования посвящен решению задачи наилучшего подбора, т. е. статистического оценивания неизвестных параметров, входящих в аналитическую запись модели, и исследованию свойств полученных оценок, их точности.

На пятом этапе (этапе верификации модели) используются различные процедуры сопоставления модельных заключений, оценок, следствий и выводов с реально наблюдаемой действительностью (этот этап называют также этапом статистического анализа адекватности модели).

Обзор и анализ мер соответствия математических моделей речевых сигналов экспериментальным данным

Для решения задач наилучшего подбора параметров и проверки на адекватность модели (четвертый и пятый этапы моделирования) необходимо строить функцию невязки (меру соответствия или ошибку модели), которая должна включать разность между расчетным значением некоторой величины, зависящей от параметров математической модели, и этой же величиной, полученной в ходе экспериментальных исследований. Функция невязки должна иметь простую математическую формулировку и желательнее быть безразмерной. Вид функции невязки определяется спецификой задачи. Адекватность (соответствие) математической модели речевого сигнала определяется путем сравнения расчетных результатов модели с экспериментальными данными (реальным речевым сигналом). При использовании математической модели (речевого сигнала) заданной функциональной зависимостью отсчетных значений модели от времени применяют обычно следующие приведенные ниже меры соответствия:

- 1) сумма квадратов ошибок (СКО);
- 2) СКО, деленная на число точек, взятых для сравнения;
- 3) взвешенная СКО;
- 4) взвешенная СКО, деленная на число точек, взятых для сравнения;
- 5) нормированная СКО;
- 6) взвешенная нормированная СКО;
- 7) корень из нормированной СКО;
- 8) сумма модулей разностей между параметрами экспериментальных данных и модели (возможны и ее модификации, вводимые по аналогии с мерами 2—7);

- 9) коэффициент множественной корреляции;
- 10) коэффициент детерминации R^2 (квадрат коэффициента множественной корреляции) [16];
- 11) коэффициент F (определяется через коэффициент детерминации) [16].

Применительно к обработке речевых сигналов из приведенных одиннадцати мер соответствия наиболее популярны следующие три (1, 5, 9). СКО или квадрат евклидовой метрики между векторами параметров экспериментальных данных $\mathbf{r}^э = (r_1^э, r_2^э, \dots, r_J^э)$ и модели $\mathbf{r}^м = (r_1^м, r_2^м, \dots, r_J^м)$ (квадрат длины вектора ошибок):

$$\varepsilon = \sum_{j=1}^J (r_j^м - r_j^э)^2, \quad (1)$$

где $r_j^э$ — значения параметра (в точках) для экспериментальных данных; $r_j^м$ — значения параметра (в точках) для математической модели; J — число точек, взятых для сравнения. Преимущества данной невязки: 1) простота записи и ясность понимания сущности ошибки; 2) гарантированное неотрицательное значение невязки; 3) данная ошибка является основой конструктивного способа получения оценок параметров математической модели — метода наименьших квадратов. Основные недостатки: 1) ε является величиной абсолютной; 2) с увеличением точек сравнения J ошибка возрастает; 3) в данной невязке каждый j -й параметр вносит одинаковый вклад в ошибку. Заметим, что очевидно при $\varepsilon \rightarrow 0$ математическая модель описывает речевой сигнал наилучшим образом.

Нормированная СКО или квадрат евклидовой метрики между векторами параметров экспериментальных данных и модели, отнесенный к квадрату евклидовой нормы вектора параметров экспериментальных данных (квадрат длины вектора ошибок нормированный на квадрат длины вектора параметров экспериментальных данных):

$$\varepsilon = \frac{\sum_{j=1}^J (r_j^м - r_j^э)}{\sum_{j=1}^J (r_j^э)^2}. \quad (2)$$

Преимущество ошибки (2) относительно (1) состоит в том, что ошибка (2) является относительной (безразмерной) величиной.

Коэффициент множественной корреляции:

$$\rho = \frac{\sum_{j=1}^J (r_j^м - \bar{r}^м)(r_j^э - \bar{r}^э)}{\sqrt{\sum_{j=1}^J (r_j^м - \bar{r}^м)^2 \sum_{j=1}^J (r_j^э - \bar{r}^э)^2}}, \quad (3)$$

где $\bar{r}^м = \frac{1}{J} \sum_{j=1}^J r_j^м$; $\bar{r}^э = \frac{1}{J} \sum_{j=1}^J r_j^э$ — математические ожидания. Следует отметить, что при $|\rho| \rightarrow 1$

наблюдаются наиболее тесные корреляционные связи и математическая модель описывает речевой сигнал наиболее адекватно.

Статистические критерии проверки на адекватность математических моделей речевых сигналов

Определение адекватности математической модели проводится только путем сравнения расчетных результатов со значениями этих же величин, полученных в эксперименте. Добиться полного совпадения этих значений в принципе невозможно. Грубо можно считать, что если расчетный результат со своим коридором неопределенности попадает в доверительный интервал экспериментального результата и этот интервал не слишком большой, то расхождение между расчетными и экспериментальными результатами носят случайный характер и расчетные результаты адекватны (соответствуют) экспериментальным. Однако для более точного количественного определения того, существенно или нет расхождение между расчетным и экспериментальным значениями, необходимо привлекать объективно-статистические критерии проверки гипотез. В качестве таких критериев проверки на адекватность математических моделей речевых сигналов можно использовать критерий Стьюдента и критерий на основе коэффициента множественной корреляции.

Критерий Стьюдента [16] позволяет ответить на вопрос о том, значимо или нет расхождение между средним значением измерений параметра $\bar{r}_j^{\text{Э}}$ и значением этой же величины $\bar{r}_j^{\text{М}}$, полученным с помощью математической модели. Недостаток критерия Стьюдента состоит в том, что в процессе выполнения экспериментов при использовании речевых сигналов не всегда можно набрать достаточно статистических данных (если число произношений речевой фразы мало или ограничено) для определения исправленной дисперсии $\sigma_j^{\text{Э}}$ и среднего значения $\bar{r}_j^{\text{Э}}$.

Критерий на базе коэффициента множественной корреляции [17] удобно использовать при сравнении расчетных (по математической модели) и экспериментальных (по реальному речевому сигналу) численных значений параметров, полученных при моделировании. Данный критерий позволяет определить, существует или нет корреляционная связь между экспериментальными $r_j^{\text{Э}}$ и расчетными $r_j^{\text{М}}$ величинами. При этом, если выполняется условие

$$|T| \geq t(\alpha; k), \quad (4)$$

то с вероятностью принятия правильной гипотезы $P = 1 - \alpha$ можно утверждать, что между ними существует "тесная" корреляционная связь. В про-

тивном случае, при $|T| < t(\alpha; k)$, гипотеза о такой связи отвергается. Наблюдаемое значение критерия имеет вид:

$$T = \frac{\rho}{\sqrt{1 - \rho^2}} \sqrt{J - 2}, \quad (5)$$

где ρ — коэффициент множественной корреляции, определяемый из соотношения (3); $t(\alpha; k)$ — критическая точка распределения Стьюдента; J — число точек, в которых проводится сравнение значений расчетных и экспериментальных величин; $k = J - 2$ — число степеней свободы.

Методика проверки на адекватность математической модели речевого сигнала

Для проверки адекватности (соответствия) математической модели речевого сигнала экспериментальным данным необходимо отработать с первого по четвертый этапы моделирования. Данные этапы предлагается проводить следующим образом:

- на первом и втором этапах моделирования проводится качественный анализ речевых сигналов, выбираются ключевые (существенные) параметры r_j , в качестве которых предлагается использовать коэффициенты корреляции;
- на третьем этапе моделирования в качестве математической модели речевого сигнала, например, предлагается использовать модель в виде импульса амплитудно-модулированного колебания с несколькими несущими частотами [9];
- на четвертом этапе моделирования рассчитывают параметры математической модели речевого сигнала, на основе метода наименьших квадратов, минимизируя функцию невязки (1).

В данной работе предлагается следующая методика проверки на адекватность (процедура верификации) математической модели речевых сигналов при использовании строгого статистического критерия на базе коэффициента множественной корреляции. Разработанная методика (пятый этап моделирования) состоит из следующих пунктов.

1. Проверка на нормальность [16] распределения коэффициентов корреляции (параметров) экспериментальных данных, для применения критерия (4). Проверка гипотезы о нормальности распределения коэффициентов корреляции речевых сигналов на основе критерия Пирсона χ^2 дала следующие результаты для исследуемых 100 реализаций речевых сигналов 10 различных дикторов при произнесении ими 10 различных парольных фраз (с участками как высокой, так и низкой вокализации): данные наблюдений согласуются с гипотезой о нормальном распределении коэффициентов корреляции с вероятностью ошибки первого рода $\alpha < 0,0005$ (при $f_d = 6000$ Гц и $J = 200$),

т. е. с вероятностью принятия правильной гипотезы $P = 1 - \alpha > 99,95\%$. При этом выполняется неравенство критерия:

$$\chi_{\text{набл}}^2 < \chi_{\text{кр}}^2(\alpha; k), \quad (6)$$

где $\chi_{\text{набл}}^2$ — наблюдаемое значение критерия; $\chi_{\text{кр}}^2(\alpha; k)$ — критическая точка распределения χ^2 ; $k = J - 3$ — число степеней свободы. Таким образом, для исследуемых реализаций речевых сигналов возможно применение статистического критерия (4).

2. Вычисление числа отсчетов коэффициентов корреляции J , достаточного для получения адекватных характеристик модели. Анализ речевых сигналов различных дикторов показал, что все значения коэффициентов корреляции заходят в доверительные границы Бартлетта [18] $\pm 3\sigma$ до 200-го отсчета для частоты дискретизации $f_d = 6000$ Гц. Таким образом, можно положить, что $J = 200$;

$$\sigma = \sqrt{\frac{1}{K} \left(1 + 2 \sum_{l=1}^L (r_l^2) \right)} \quad (7)$$

— стандартная ошибка для коэффициентов корреляции [18]; K — число отсчетов речевого сигнала; L — некоторое число отсчетов коэффициентов корреляции, не превышающее примерно $K/4$ [18].

Следует отметить, что если частоту дискретизации взять не $f_{d1} = 6000$ Гц, а другую, f_{d2} , необходимо учитывать, что значение $J_1 \equiv J(f_{d1})$ также изменится на $J_2 \equiv J(f_{d2})$. Если обозначить как $\tau_{3\sigma}$ — время, начиная с которого все отсчеты коэффициентов корреляции заходят в границы Бартлетта $\pm 3\sigma$ (7), то можно записать

$$J \equiv J(f_d) = \tau_{3\sigma} / \Delta = \tau_{3\sigma} f_d, \quad (8)$$

где $\Delta = 1/f_d$ — интервал дискретизации. Как показывают экспериментальные исследования, значение $\tau_{3\sigma}$ практически не зависит от f_d , поэтому из соотношения (8) следует, что

$$J_2 = \frac{f_{d2}}{f_{d1}} J_1. \quad (9)$$

3. Проводится сравнение экспериментальных и расчетных по модели параметров речевого сигнала с привлечением статистического критерия проверки гипотез на базе коэффициента множественной корреляции (4). При этом:

а) задается уровень значимости (вероятность ошибки первого рода) α (как правило, α выбирают равным 0,05; 0,01; 0,005 или 0,001 в зависимости от требуемой точности соответствия);

б) задается число сравниваемых коэффициентов корреляции J (для определенной частоты дискретизации f_d);

в) вычисляется коэффициент множественной корреляции ρ по формуле (3);

г) вычисляется наблюдаемое значение критерия T по соотношению (5);

д) вычисляется критическая точка распределения Стьюдента $t(\alpha; k)$ для числа степеней свободы $k = J - 2$;

е) проверяется выполнение условия (4).

Если условие (4) выполнено, то с вероятностью принятия правильной гипотезы $P = 1 - \alpha$ можно утверждать, что между параметрами (коэффициентами корреляции) речевого сигнала, рассчитанными по экспериментальным данным $r_j^{\text{Э}}$ и по математической модели $r_j^{\text{М}}$, существует "тесная" корреляционная связь. То есть математическая модель речевого сигнала адекватна (соответствует) экспериментальным данным с вероятностью P .

Результаты

На основе проведенных исследований можно сделать следующие рекомендации практического использования трех базисных приведенных мер соответствия модели.

Невязку (1) — сумму квадратов ошибок — в большинстве случаев удобно применять для аналитических выводов значений оценок параметров математической модели.

Ошибку модели (2) — нормированную сумму квадратов ошибок — целесообразно использовать для сравнительного анализа ошибок, особенно в случае применения процентов.

Использование меры соответствия в виде коэффициента множественной корреляции (3) необходимо применять для более точного количественного определения степени адекватности модели на основе статистического анализа при использовании критерия на базе коэффициента множественной корреляции (4).

Результаты применения указанных мер соответствия математической модели [9] реальному речевому сигналу (содержащему вокализованные участки речи) дали следующие результаты при использовании критерия на базе коэффициента множественной корреляции (4). Математическую модель [9], на основе характеристики в виде коэффициентов корреляции r_j речевого сигнала (для параметров $f_d = 6000$ Гц; $J = 200$), можно считать адекватной реальному речевому сигналу с вероятностью принятия правильной гипотезы $P = 1 - \alpha > 99,9\%$.

Выводы. Таким образом, дан обзор и выполнена систематизация мер соответствия математической

модели речевого сигнала экспериментальным данным, даны рекомендации применения двух основных мер соответствия модели. Разработанная методика проверки математической модели речевого сигнала на адекватность при использовании статистического критерия проверки гипотез на базе коэффициента множественной корреляции позволяет корректно проводить верификацию модели.

Список литературы

1. **Сорокин В. Н.** Фундаментальные исследования речи и прикладные задачи речевых технологий // Речевые технологии. 2008. № 1. С. 18—48.
2. **Назаров М. В., Прохоров Ю. Н.** Методы цифровой обработки и передачи речевых сигналов. М.: Радио и связь, 1985. 176 с.
3. **Рабинер Л. Р., Шафер Р. В.** Цифровая обработка речевых сигналов. М.: Радио и связь, 1981. 496 с.
4. **Голубинский А. Н.** К вопросу о разработке математической модели речевого сигнала на основе особенностей его характеристик // Системы управления и информационные технологии. 2008. № 2. С. 80—85.
5. **Маркел Дж., Грей А. Х.** Линейное предсказание речи. — М.: Связь, 1980. 308 с.
6. **Ролдугин С. В., Голубинский А. Н., Вольская Т. А.** Модели речевых сигналов для идентификации личности по голосу // Радиотехника. 2002. № 11. С. 79—81.
7. **Styllianou Y.** Apply the harmonic plus noise model in concatenative speech synthesis // IEEE Trans. on Speech and Audio Process. 2001. Vol. 9. N 1. P. 21—29.

8. **Zavarehei E., Vaseghi S., Yan Q.** Noisy speech enhancement using harmonic-noise model and codebook-based post-processing // IEEE Trans. on Speech and Audio Process. 2007. Vol. 15. N 4. P. 1194—1203.
9. **Голубинский А. Н.** Разработка математической модели речевого сигнала в виде импульса АМ-колебания с несколькими несущими частотами, применительно к задаче верификации личности по голосу / А. Н. Голубинский; Воронежский институт МВД России. — Воронеж, 2008. — 29 с. — Деп. в ВИНТИ 09.07.08, № 591-В2008.
10. **Якушев Д. И., Скляров О. П.** Моделирование гласных звуков // Акустический журнал. 2003. Т. 49. № 4. С. 567—569.
11. **Фант Г.** Акустическая теория речеобразования. М.: Наука, 1964. 284 с.
12. **Голубинский А. Н.** Физическая интерпретация математической модели речевого сигнала, в виде импульса АМ-колебания с несколькими несущими частотами, на основе принципов речеобразования // Системы управления и информационные технологии. 2008. № 3. С. 78—82.
13. **Маркин Д. Н., Уваров В. К.** Результаты новых исследований на основе модуляционной теории звуковых сигналов // В сб. Факультету аудиовизуальной техники — 75 лет. — СПб.: Изд. СПбГУКиП, 2005. С. 36—41.
14. **Голубинский А. Н.** Обработка речевого сигнала на основе модели в виде импульса АМ-колебания с несколькими несущими частотами // Телекоммуникации. 2008. № 12. С. 13—17.
15. **Айвазян С. А., Енюков И. С., Мешалкин Л. Д.** Прикладная статистика: Основы моделирования и первичная обработка данных. — М.: Финансы и статистика, 1983. 471 с.
16. **Кендалл М. Дж., Стюарт А.** Статистические выводы и связи. М.: Наука, 1973. 899 с.
17. **Айвазян С. А., Енюков И. С., Мешалкин Л. Д.** Прикладная статистика: Исследование зависимостей. М.: Финансы и статистика, 1985. — 487 с.
18. **Бокс Дж., Дженкинс Г.** Анализ временных рядов. Прогноз и управление. — М.: Мир, 1974. Вып. 2. 408 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

УДК 61:004.62

В. С. Анищенко, д-р физ-мат. наук, проф.,
Саратовский государственный университет им. Н. Г. Чернышевского,
Т. И. Булдакова, д-р техн. наук, проф.,
Поволжская академия государственной службы имени П. А. Столыпина,
П. Я. Довгалецкий, д-р мед. наук, проф., Саратовский НИИ кардиологии,
В. Б. Лифшиц, д-р мед. наук, проф., Саратовский государственный медицинский университет,
В. И. Гриднев, канд. мед. наук, Саратовский НИИ кардиологии,
С. И. Суягинов, канд. техн. наук, доц., Саратовский государственный технический университет,
e-mail: okb_sstu@sstu.ru

Концептуальная модель виртуального центра охраны здоровья населения

Рассмотрены основные положения, принципы, условия и механизмы реализации в форме виртуального центра новой концепции в здравоохранении — персонализированной медицины. Предложен модельный подход к оценке функционального состояния человека на основе структурно-параметрической идентификации биосистемы "сердце — сосуды — легкие".

Ключевые слова: персонализированная медицина, мониторинг, оценка функционального состояния, виртуальный физиологический образ, биосигналы, система "сердце — сосуды — легкие", модельные уравнения.

Введение

Важнейшим обязательством государства является охрана здоровья населения как наивысшей индивидуальной и общественной ценности. Однако практикуемая методология здравоохранения, основанная на использо-

вании среднестатистических показателей патологии/норма и стандартных схем лечения, является малоэффективной. Вследствие зарегулированности, использования обобщенных методик диагностики состояния она не может обеспечить не только индивидуальное рас-

познавание ранних форм заболеваний и решения различных задач прогнозирования, но зачастую не обеспечивает эффективность и безопасность лечения.

Эта проблема характерна не только для России, но и для других стран. Поэтому в Европейских странах предложена и развивается новая концепция системы охраны здоровья населения. Согласно ей, основные задачи охраны здоровья лежат не в плоскости высокотехнологичных методов лечения, а фокусируются на первичной профилактике и ранней диагностике [1]. Приоритетной составляющей новой концепции становятся регулярный мониторинг и оценка состояния здоровья населения, персонализация здравоохранения, оказание медицинской помощи с учетом индивидуальных особенностей пациента.

Новая концепция здравоохранения и ее реализация с использованием современных информационных технологий позволяют сформировать в России систему качественного и доступного здравоохранения, минимизировав при этом расходы государства.

Задачи персонализированного здравоохранения

Традиционные системы оказания помощи пациентам базируются на предоставлении медицинских услуг только в случае возникновения заболевания и непосредственного обращения к врачу. Такие системы основаны на сети лечебно-профилактических учреждений (ЛПУ), где имеется необходимое медицинское оборудование. Они не отвечают требованиям оперативности, мобильного мониторинга и диспансеризации, учета индивидуальных особенностей пациента.

Развитие технологий "*sensor-on-a-chip*" и открытых телекоммуникационных систем позволяют приступить к реализации методологии "госпиталь на дому" ("*hospital-at-home*") [2, 3]. Основным принципом данной методологии является мониторинг функционального состояния и оказание медицинской помощи пациентам независимо от их местоположения, в том числе и в домашних условиях. При этом источниками объективной информации о функциональном состоянии пациента являются мобильные измерительные системы, выполненные по технологиям "*sensor-on-a-chip*" и "*laboratory-on-a-chip*". Например, телекоммуникационная система регистрации биосигналов, вмонтированная в нательную одежду, позволяет вести перманентный мониторинг состояния человека, не ограничивая его мобильности [4]. Физическое местоположение пациента в этом случае не имеет значения при определении его функционального состояния. Биосигналы, регистрируемые встроенными датчиками, передаются по каналам связи в медицинские центры мониторинга и обработки информации, используя общую информационную инфраструктуру здравоохранения [5].

Математизация медицины [6] создает основу для разработки новых системно-информационных технологий персонального мониторинга, диагностики и прогнозирования состояния отдельного пациента. В этом направлении важной проблемой является создание виртуальной физиологии человека. Виртуальная "копия" пациента, построенная посредством математических моделей элементов и подсистем организма, описывает деятельность физиологических подсистем человека и представляет собой его виртуальный физиологический образ (ВФО) [7]. На основе таких моделей медицинское обслуживание переходит на качественно новый рубеж персонализированного и упреждающего здравоохранения, основанного на

технологиях моделирования и имитации физиологии человека и процессов, связанных с заболеваниями.

Учитывая индивидуальные особенности индивидуума в форме математических моделей, персонализированное здравоохранение решает следующие задачи: диагностику на ранних стадиях и прогнозирование; повышение эффективности лечения за счет подбора медикаментов и схем лечения, учитывающих специфические свойства организма конкретного пациента; повышение безопасности лечения за счет оптимизации медицинского вмешательства и предотвращения ошибок; информирование и привлечение самих пациентов к более ответственной заботе о своем здоровье; интеграцию разнородной информации и извлечение новых знаний, внедрение новых научных знаний в медицинскую практику.

Информационно-телекоммуникационные технологии, математизация и другие формы представления знаний позволяют эффективно реализовать систему персонализированного здравоохранения в форме виртуальной организации [8].

Виртуальный центр объединяет на базе общего информационного пространства все составляющие элементы системы здравоохранения, обеспечивая сбор более полной информации, глубокий анализ и обмен большими объемами данных, в значительной степени автоматизируя труд медицинского персонала. Он предоставляет работникам в сфере здравоохранения удобный инструмент реализации функций хранения, накопления и обработки данных о пациентах на протяжении всей их жизни (рис. 1).

Основной проблемой организации информационного обмена является информационная совместимость. Представляется, что естественным решением указанной проблемы может быть использование CALS-технологий. Подтверждением этого является, например, широкое использование языка XML для представления, хранения и использования различными пользователями биотехнических и физиологических данных. Кроме того, форматы описания данных (в частности биосигналов), принятые в медицине, сходны со структурой передаваемого пакета данных, предписанной стандартом MHL 1840.

Ключевым понятием CALS-технологий является понятие Единого информационного пространства. На рис. 2 представлены основные источники и потребители информации в структуре виртуального центра охраны здоровья.

Поставщик ресурсов (провайдер) виртуального центра представляет собой совокупность устройств и приложений, которые управляют и обрабатывают персональную информацию, выявляют скрытые закономер-



Рис. 1. Жизненный цикл ВФО

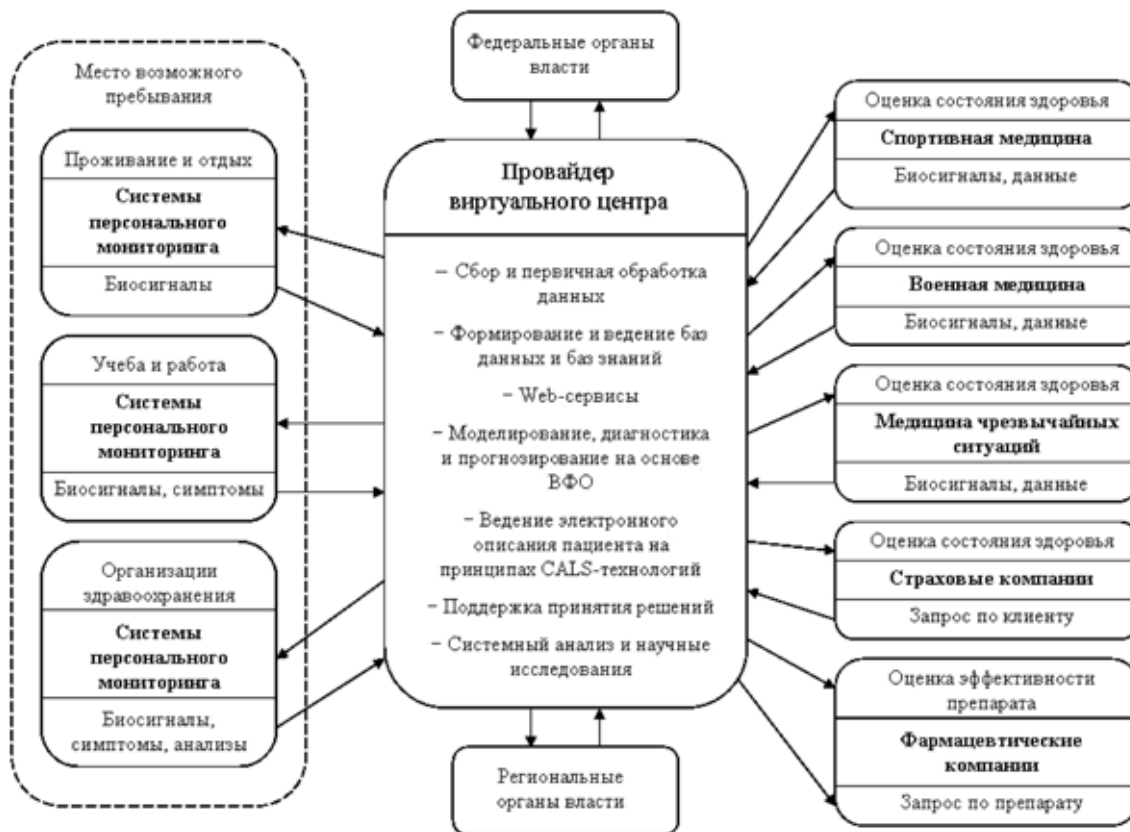


Рис. 2. Информационное пространство виртуального центра

ности и помогают врачам и пациентам. Учитывая большое число пользователей, необходимость интеграции разнородных информационных систем, представляется обоснованным использовать методологию сервис-ориентированной архитектуры для построения программной инфраструктуры виртуального центра.

Переход к виртуальной форме организации всего здравоохранения предполагает поэтапную реализацию отдельных проектов. Приоритетной задачей первого этапа является создание методов и технических средств экспресс-мониторинга и оценки состояния здоровья человека.

Показатели здоровья и методы оценки функционального состояния организма

В конце прошлого века возникла и получила признание новая научная концепция оценки уровня здоровья и прогнозирования функциональных состояний организма. В соответствии с этой концепцией в качестве меры здоровья принимается способность организма приспосабливаться, адаптироваться к окружающей среде [9]. В случае снижения адаптационных возможностей организма возникает период перехода от здоровья к болезни. Эти переходные состояния организма получили название *донозологических* [10]. Поэтому донозологическая диагностика, позволяющая определить изменение состояния организма до проявления патологий, является основой для развития упреждающей медицины. Находясь в донозологических состояниях, организм приспосабливается к внешним воздействиям и новым условиям путем изменения уровня функционирования отдельных органов и подсистем. Это происходит за счет работы регуляторных механизмов, направленных на мобилизацию

функциональных резервов. Именно степень напряжения регуляторных систем, необходимая для сохранения гомеостаза, определяет текущее функциональное состояние человека. Задачей донозологической диагностики является распознавание текущего функционального состояния по показателям степени напряжения регуляторных механизмов и функциональных резервов. Таким образом определяются адаптационные возможности организма и дается оценка состояния здоровья человека.

Действие регуляторных систем связано, в первую очередь, с изменением энергетического обмена и соответствующей потребности организма в расходе кислорода. Поэтому в настоящее время состояние всего организма оценивают по состоянию сердечно-сосудистой системы, которая осуществляет транспорт кислорода ко всем органам.

Сейчас уже общепризнано, что вариабельность или разброс физиологических параметров является интегральным методом оценки состояния механизмов регуляции физиологических функций в организме человека. В настоящее время используются различные методы анализа вариабельности биосигналов сердечно-сосудистой системы. Все они основаны на математическом анализе спектральных и временных характеристик временных рядов — дискретной эволюции биосигналов во времени.

Методология оценки функционального состояния по вариабельности основывается на представлении регуляции в организме в форме непрерывных процессов взаимодействия между управляющими и управляемыми элементами функциональной системы. Возникающие при этом колебания значений физиологических параметров отражают деятельность механизмов управления, и, та-

ким образом, информация о состоянии различных звеньев системы управления заключена в "функциях разброса" [11]. Вариации физиологических параметров приводят к вариациям различных параметров регистрируемых биосигналов.

Таким образом, существующие методы позволяют оценить "функции разброса" параметров по косвенной информации, содержащейся в биосигналах.

Однако представляется более перспективным строить математические модели биосистем на основе системного подхода и оценивать разброс параметров этих моделей в процессе адаптации организма.

Системный подход к оценке адаптационных способностей организма

С позиций системного подхода к исследованию сложных систем [12] важно из общей организации биосистемы выделить контур, отражающий функциональное состояние всего организма, определить модели управляющих и управляемых элементов, описать взаимосвязи между ними.

Здесь возникают следующие задачи: выбор биосистем, деятельность которых отражает функциональное состояние всего организма; разработка структуры (структурная идентификация) моделей входящих в нее элементов; идентификация параметров этих моделей.

Следует отметить, что сердечно-сосудистая система выполняет только функции транспорта кислорода, поэтому логично рассматривать для оценки функционального состояния организма всю систему кислородного обеспечения организма. Кратко назовем эту систему "сердце — сосуды — легкие".

Система "сердце — сосуды — легкие" с ее многоуровневой регуляцией представляет собой функциональную систему, конечным результатом деятельности которой является обеспечение заданного уровня функционирования целостного организма. Обладая сложными нервно-рефлекторными и нейрогуморальными механизмами, система оксигенации обеспечивает своевременное адекватное обеспечение кислородом различных структур. При прочих равных условиях можно считать, что любому заданному уровню функционирования целостного организма соответствует эквивалентный уровень функционирования аппарата оксигенации.

Оценка и прогнозирование функционального состояния целостного организма по данным исследования системы "сердце — сосуды — легкие" основывается на следующих положениях: изменения динамических процессов в биосистеме "сердце — сосуды — легкие" возникают раньше, чем соответствующие функциональные нарушения; изменения параметров взаимосвязей в механизме регуляции рассматриваемой биосистемы происходят раньше, чем появляются патологические изменения; исследование процессов адаптации, временной организации и координации биофизических процессов в системе "сердце — сосуды — легкие" позволяет выявлять самые начальные изменения в управляющем звене целостного организма.

Особенность рассматриваемой системы заключается в том, что все ее элементы совершают механические движения и, следовательно, подчиняются законам механики. Основываясь на концепции иерархии упрощенных моделей и мягком моделировании, будем строить приближенные функционально-структурные модели, отражающие основные особенности динамики элементов [13].

Можно сформулировать следующие требования к моделям первого иерархического уровня: непротиворечивость законам биофизики; качественное отображение основных процессов и эффектов; соответствие выбранным на данном уровне критериям адекватности.

В качестве исходной структуры той или иной модели предлагается выбирать наиболее адекватную из ряда базовых моделей, описывающих нелинейные колебания, или их модификации. Параметрическая идентификация выбранных моделей осуществляется по совокупности зарегистрированных биосигналов. В качестве исходных могут использоваться, например, ЭКГ, кинетокардиограмма, сфигмограмма сонной артерии, пневмотахограмма, спирограмма и др. Это те биосигналы, структура которых определяется параметрами генерирующих их элементов рассматриваемой биосистемы. Возможность представления сложных биофизических процессов относительно простыми модельными уравнениями обуславливается следующими обстоятельствами.

1. С позиций синергетики рассматриваемая биосистема имеет очень много степеней свободы. Однако в процессе естественной эволюции выделяются несколько главных степеней, к которым подстраиваются все остальные. Их динамика описывается сравнительно простыми нелинейными уравнениями невысокого порядка и отражает основные свойства всей сложной системы [14].

2. Рассматриваются процессы вполне определенного временного масштаба, обусловленного периодами колебаний.

3. Системы, функционирующие в режиме автоколебаний, имеют более простое описание.

Реконструированные математические модели содержат явную и скрытую информацию об исследуемой биосистеме. Поэтому данный подход позволяет выполнить анализ временной организации физиологического процесса и/или оценить функциональное состояние подсистемы.

Доступность системы электрофизиологических биосигналов "сердце — сосуды — легкие" для измерения в сочетании с модельным представлением биосигналов позволяет выполнять скрининг-диагностику функционального состояния организма и решать задачи прогнозирования. Кроме того, вычисление показателей взаимосвязей в моделях исследуемых подсистем на основе зарегистрированных биосигналов позволяет определить характеристики или признаки изменения функционального состояния на ранних (донозологических) стадиях. Предлагаемый подход открывает принципиально новую возможность не только диагностики, но и описания физиологии и представления знаний в медицинских информационных системах.

В работах [15, 16] приведены примеры идентификации функциональных взаимосвязей сердечно-сосудистой системы (системы "сердце — сосуды") на основе анализа синхронно регистрируемых биосигналов перемещений стенки кровеносного сосуда x и электрической активности сердца e . Например, в работе [16] модель подсистемы "сердце — сосуды" описывается обобщенным уравнением Ван-дер-Поля—Дуффинга:

$$\ddot{x} + [\varepsilon_1(x^2 - r^2) + \varepsilon_2(\dot{x}^2 - w_0^2 \cdot r^2)]\dot{x} + ax^3 = F_{\text{связи}},$$

где $w = \sqrt{a}$ — частота основного гармонического колебания стенки сосуда; $F_{\text{связи}}$ — функция, описывающая воздействие сердечной активности на динамику стенки сосуда; $\varepsilon_1, \varepsilon_2, w_0, r_0$ — неизвестные параметры, которые

находятся на этапе параметрической идентификации модели. Значения $\ddot{x}(t)$, $\dot{x}(t)$, $x(t)$, $\dot{e}(t)$, $e(t)$ определяются по экспериментально полученным данным. Функция связи $F_{\text{связи}}$ реализована на нейронной сети Вольтерра.

Весовые коэффициенты функции связи могут рассматриваться как дополнительный диагностический критерий при диагностике функционального состояния человека.

Отметим, что структура модели отображает общие, устойчивые свойства биосистемы, а параметры моделей, которые определяются по биосигналам, отображают индивидуальные, специфические свойства. Поэтому предложенное модельное представление биосистемы носит сугубо индивидуальный характер, присущий конкретному человеку.

Это создает предпосылки для перехода от концепции среднестатистического больного к персонализированной медицине. Рассматриваемый подход позволяет создавать структурно-параметрические модели, объективно отражающие состояние органа при действии различных факторов, в удобной форме хранить эту информацию и легко обрабатывать ее различными математическими методами.

Принципы компьютерного представления ВФО

Составной частью общей виртуальной модели физиологии человека является компьютерная модель биосистемы "сердце — сосуды — легкие". Принципы ее описания и представления могут быть использованы при разработке других моделей биосистем.

Реализация компьютерной модели в форме программного модуля осуществляется на принципах переносимости, многократного использования кода, модульности, независимости от программно-аппаратных платформ и объектно-ориентированного подхода.

Программная модель ВФО имеет несколько уровней представления, где реализуются:

- математические модели физиологических процессов;
- управление процессом идентификации и моделирования при решении различных прикладных задач, например прогнозирование изменения функционального состояния при тех или иных воздействиях;
- интерфейсы взаимодействия с базой данных биосигналов и модулями обработки.

Модельные уравнения описывают динамику отдельных элементов биосистемы и механизмы связей между ними. Возможные варианты структур моделей описываются с использованием языка XML и хранятся в соответствующей базе данных. Математические модели, реализованные подобным образом, могут использоваться многократно в различных исследованиях многими пользователями совместно с другими моделями. По существу, это есть составляющая электронного описания изделия в терминологии CALS-технологий.

Преобразование ВФО в программный модуль может быть выполнено различными способами, например с промежуточным описанием уравнений в декларативной форме с использованием языка *Modelica*.

Далее формируются классы объектов на принципах объектно-ориентированного программирования. И наконец, объединяя класс модели ВФО, класс управления процессом и несколько классов источников данных, строится класс ВФО.

На заключительном этапе получения автономного программного модуля осуществляется параметрическая

идентификация виртуальной модели с использованием биосигналов конкретного пациента. В результате выполняется "привязка" модели к конкретному человеку.

Прототип виртуального центра охраны здоровья

Основные положения концепции, предложенная модельная форма представления знаний о функциональном состоянии человека реализованы в биомедицинской информационно-аналитической системе (БИАС) [17]. Функциями БИАС являются:

- регистрация входных данных (электрокардиограмма, пульсограмма, пневмотахограмма);
- формирование моделей подсистем по биосигналам;
- исследование и формирование классификационных признаков;
- генерация отчетов о функциональном состоянии организма пациента;
- учет и хранение персональной информации (личных данных пациентов, информации о госпитализации, истории болезни, результатов проведенных исследований).

При разработке рассматриваемого макетного варианта виртуального центра акцент был сделан на использование открытого программного обеспечения.

Макет виртуального центра представляет собой распределенную БИАС, содержащую устройства удаленного доступа, web-сервер, клиентские приложения и сервер обработки (рис. 3).

Устройства удаленного доступа регистрируют электрофизиологические биосигналы и другую информацию о пациенте и отсылают их подсистеме обработки информации, используя различные протоколы и каналы связи.

В системе применяются два типа устройств удаленного доступа. Первый тип — мобильный терминал (MT) регистрации биосигналов на основе микроконтроллера



Рис. 3. Структура прототипа

[18]. Второй тип — это информационно-измерительная система ЛПУ с выходом в Интернет.

Клиентское приложение — это пользовательский интерфейс, с помощью которого осуществляется доступ к ресурсам виртуального центра. Представленная БИАС предусматривает работу двух групп пользователей: исследователей и лечащих врачей. Для каждой группы пользователей определен набор системных функций и их комбинации.

Наконец, сервер обработки информации выполняет функции провайдера виртуального центра. Это набор программно-аппаратных средств, осуществляющих обработку и накопление знаний о пациентах. В основу данной подсистемы положена технология web-сервисов, оперирующая тремя основными понятиями: набор сервисов, генератор сценариев и исполняемый сценарий. Фактически каждый исполняемый сценарий является приложением, разработанным для определенной группы пользователей. Набор сервисов представляет собой совокупность базовых функций, которые когда-либо могут потребоваться для работы приложений. Логика формирования сценариев основывается на predetermined правилах, адекватных конкретному случаю.

Следует отметить, что для универсального использования баз данных (с медицинской информацией) несколькими приложениями был реализован инструмент стандартизации данных. В основу инфраструктуры БИАС была положена открытая XML-технология, являющаяся удобным форматом представления, преобразования и обработки биомедицинских данных.

Заключение

Развитие микроэлектроники и телекоммуникаций позволяют включить пациента в единое информационное пространство системы здравоохранения, независимо от его местоположения. Проблемы информационной совместимости предлагается решать на принципах CALS-технологий, что позволяет объединить в информационном пространстве ЛПУ и предприятия медицинской промышленности. Сервис-ориентированная архитектура является основой программной реализации виртуального центра.

Базовой составляющей "электронного описания изделия" логично принять виртуальную физиологию человека, включающую математическую модель биосистемы "сердце — сосуды — легкие". Использование языка XML делает модель доступной для различных групп пользователей. Эта модель является удобной формой представления знания и позволяет отслеживать состояние здоровья человека на протяжении всего жизненного цикла.

Помимо поддержки клинической медицины, накопленные о пациенте данные, представленные в унифицированной форме электронного описания (расширенной форме электронной болезни), могут быть использованы в научных целях для выявления новых закономерностей.

Программно-аппаратная часть системы строится на принципах открытой архитектуры и свободно распространяемого ПО, что позволяет обеспечить гибкую перенастройку, интеграцию разнородных данных, моделей и процессов.

Представляется обоснованной как с технической, так и с финансовой точек зрения идея реализации виртуаль-

ного центра совместно с одной из телекоммуникационных компаний.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 07-07-12066).

Список литературы

1. **Дмитриева Н. В., Глазачев О. С.** Индивидуальное здоровье и полипараметрическая диагностика функциональных состояний организма (системно-информационный подход). М.: Медицина, 2000. 214 с.
2. **Williams G., Doughty K., Bradley D.** A Systems Approach to Achieving CarerNet — An Integrated and Intelligent Telecare System // IEEE Transactions on Information Technology in Biomedicine. 1998. Vol. 2, N 1. P. 1—9.
3. **Winters J., Wang Y.** Wearable Sensors and Telerehabilitation // IEEE Engineering in Medicine and Biology Magazine. 2003. N 3. P. 56—65.
4. **Paradiso R., Loriga G., Taccini N.** A Wearable Health Care System Based on Knitted Integrated Sensors // IEEE Transactions on Information Technology in Biomedicine. 2005. Vol. 9. N 3. P. 337—344.
5. **Гончаров Н. Г., Гулиев Я. И., Гуляев Ю. В.** и др. Вопросы создания Единого информационного пространства в системе здравоохранения РАН // Информационные технологии и вычислительные системы. 2006. № 4. С. 83—94.
6. **Богомолов А. В., Гридин Л. А., Кукушкин Ю. А., Ушаков И. Б.** Диагностика состояния человека: математические подходы. М.: Медицина, 2003. 464 с.
7. **Prado M., Roa L., Reina-Tosina J.** Virtual Center for Renal Support: Technological Approach to Patient Physiological Image // IEEE Transaction on biomedical engineering. 2002. Vol. 49. N 12. P. 1420—1430.
8. **Уорнер М., Витцель М.** Виртуальные организации. Новая форма ведения бизнеса в XXI веке / Пер. с англ. М.: Добрая книга, 2005. 296 с.
9. **Баевский Р. М., Берсенева А. П.** Оценка адаптационных возможностей организма и риск развития заболеваний. М.: Медицина, 1997. 236 с.
10. **Баевский Р. М., Казначеев В. П.** Диагноз донозологический. М.: БМЭ, 1978. Т. 7. С. 253—255.
11. **Баевский Р. М.** Проблема оценки и прогнозирования функционального состояния организма и ее развитие в космической медицине // Успехи физиологических наук. 2006. Т. 37, № 3. С. 42—57.
12. **Прангишвили И. В.** Системный подход и общесистемные закономерности. Сер. Системы и проблемы управления. М.: Синтез, 2000. 528 с.
13. **Малинецкий Г. Г.** Математические основы синергетики: Хаос, структуры, вычислительный эксперимент. Изд. 5-е — Изд-во ЛКИ, 2007. 312 с. (Синергетика: от прошлого к будущему).
14. **Малинецкий Г. Г., Потапов А. Б.** Современные проблемы нелинейной динамики. М.: Эдиториал УРСС, 2000. 336 с.
15. **Коблов А. В., Ланцберг А. В., Самочетова Н. С., Суятинов С. И.** // Вестник СГТУ. 2007. № 4 (27). С. 31—37.
16. **Булдакова Т. И., Лифшиц В. Б., Суятинов С. И.** Выявление групп риска у людей с высоким уровнем холестерина: статистический и модельный подходы // Информационные технологии моделирования и управления. 2008. № 4 (47). С. 363—368.
17. **Булдакова Т. И., Гриднев В. И., Кириллов К. И.** и др. Программно-аналитический комплекс модельной обработки биосигналов. // Биомедицинские технологии и радиоэлектроника. 2009. № 1. С. 53—57.
18. **Булдакова Т. И., Коблов А. В., Суятинов С. И.** Информационно-измерительный комплекс совместной регистрации и обработки биосигналов // Приборы и системы. Управление, контроль, диагностика. 2008. № 6. С. 41—46.

С. Д. Махортов, канд. физ.-мат. наук, доц.,
Воронежский государственный университет,
e-mail: sd@expert.vrn.ru

Интегрированная среда логического программирования LPExpert

В ряде ранее опубликованных работ автором был предложен алгебраический аппарат (теория LP-структур), позволяющий формализовать продукционно-логический вывод на основе теории решеток и отношений. Этот аппарат применим для автоматизированного исследования свойств баз знаний на основе формальных моделей, включая их верификацию и оптимизацию. В настоящей статье кратко представлены функциональные возможности созданного автором пакета программ LPExpert, предназначенного для разработки продукционных систем и содержащего компьютерную реализацию LP-структур.

Ключевые слова: продукционная система, алгебраическая модель, управление знаниями, компьютерная реализация.

Введение

Теоретическую основу построения пакета LPExpert составили концептуальные положения монографии [1]. Несмотря на то, что эта книга носит в большей мере учебный характер, ее идеи оказались продуктивными и в дальнейшем развивались в ряде работ по экспертным системам (ЭС), в первую очередь обучающим (например, [2–3]). Пакет программ LPExpert реализован в Delphi 5. В последнее время к его функциональности была добавлена *dll*-библиотека — интерфейс класса *LPStructure* (разработанного в *MS VC++*), что значительно расширило возможности пакета в плане эффективности его применения для создания баз знаний.

1. Структура базы знаний

Опишем структуру базы знаний (БЗ) пакета *LPExpert*. Она представляет собой расширение структур БЗ, принятых в работе [2] и впоследствии в работе [3].

БЗ содержит знания эксперта в некоторой предметной области. Она включает известные факты, выраженные в виде объектов и их значений. В *LPExpert* принята запись факта в виде пары "объект = значение", причем данное утверждение может быть верным с некоторой вероятностью, понятие которой заменено *коэффициентом доверия* — целым числом от 1 до 100. Некоторые

объекты могут быть многозначными, т. е. одновременно допускают несколько значений. С ними в БЗ связаны списки значений.

С точки зрения ЭС объекты можно разделить на следующие три класса.

1. Объекты, значения которых невозможно вывести с помощью правил продукционной системы из значений других объектов. Система может получить их лишь непосредственно от пользователя или извлечь из имеющихся фактов. Назовем эти объекты *начальными*.

2. Объекты, значения которых выводятся с помощью правил, причем эти объекты имеют практический смысл для пользователя. Будем называть их *объектами экспертизы*.

3. Объекты, значения которых можно вывести, однако они недоступны пользователю и созданы разработчиком БЗ для улучшения ее структуры. Такие объекты называются *внутренними*.

Во время считывания (компиляции) БЗ система автоматически проводит классификацию объектов.

Для каждого начального объекта в БЗ обычно заготовлен вопрос, который может быть задан пользователю в целях получения значения. С такими объектами также связано понятие *разрешенных значений*, которые составляют области их определения. При ответе на вопрос пользователь выбирает значение из предлагаемого списка, составленного из разрешенных значений.

Важной составной частью БЗ является совокупность правил (продукций), с помощью которых система делает выводы о значениях объектов. Правило состоит из предпосылки и заключения, которые представляют собой подмножества фактов БЗ. Каждый факт заключения при этом может иметь коэффициент доверия, а при его отсутствии факт считается достоверным.

Таким образом, БЗ содержит следующую информацию:

- 1) список объектов экспертизы;
- 2) список многозначных объектов;
- 3) список разрешенных значений для каждого начального объекта;
- 4) вопросы, соответствующие начальным объектам;
- 5) продукционные правила;
- 6) накопленные факты — рабочее множество.

Кроме того, элементы БЗ могут содержать дополнительную информацию (комментарии, звуки, изображения), которые делают интерфейс более дружелюбным.

Расширения возможностей пакета *LPExpert* основаны на представлении наборов фактов и правил базы знаний *LP*-структурой [4]. Каждый единственный факт изображается точкой решетки — булеана, предпосылка и заключение правила — двумя элементами решетки, а сами правила представляются парами бинарного отношения в *LP*-структуре.

Хранение БЗ организовано в виде набора текстовых файлов. Для простоты будем называть их файлами правил. База знаний некоторой предметной области может располагаться в нескольких файлах правил, их перечень хранится в файле проекта. Синтаксис БЗ прост и интуитивно понятен.

2. Структура пакета LPExpert и принципы реализации

Пакет состоит из следующих основных блоков: редактор, компилятор, машина вывода, модуль интерфейса класса *LPStructure*. Значительную часть в его архитектуре занимает также интерфейс пользователя.

Редактор предназначен для подготовки файлов БЗ в текстовом режиме. Он предоставляет пользователю все основные функции стандартного текстового редактора. Еще одна из существенных функциональных возможностей редактора — взаимодействие с компилятором и выделение текста, содержащего ошибку.

Другая группа возможностей редактора связана с просмотром хранящейся в памяти двоичной информации — откомпилированной БЗ в виде объектов, значений и правил. Это оказывает существенную помощь при отладке БЗ.

Компилятор содержит блоки лексического и синтаксического анализа. При запуске на компиляцию БЗ считывается в память и переводится в двоичное представление. Если во время компиляции обнаруживается ошибка, открывается соответствующий исходный файл в редакторе, в нем выделяется содержащая ее строка и пользователю сообщается о характере ошибки.

Двоичное хранение БЗ реализуется путем широкого использования динамических структур, в основном, коротких строк и индексированных списков. Следует отметить, что применение таблиц имен, и соответственно, использование индексов вместо самих имен позволило значительно уменьшить объем занимаемой памяти по сравнению с реализацией, предложенной в работе [2]. Высокая скорость считывания множества правил при этом практически сохранена.

Ядром ЭС является *машина вывода*, которая способна интерпретировать правила, используя факты БЗ, для решения поставленной задачи, а именно, определения значений заданного объекта экспертизы. Этот процесс называется *экспертизой*. Она осуществляется на основе чистого обратного вывода — путем формулирования пробных гипотез и их рекурсивной проверки на соответствие указанной цели. Основной блок машины вывода реализован в виде рекурсивной процедуры.

Существенной особенностью машины вывода в *LPExpert* является возможность для пользователя в очередном диалоге отказаться от выбора значения объекта, и вместо этого сделать шаг назад к предыдущему диалогу. Во время вывода на экран опционально выдается трассировка применяемых правил. Таким образом, имеются эффективные средства отладки разрабатываемых БЗ.

Реализация указанной выше отмены выбора не является операцией тривиальной. Причины в том, что

после сделанного пользователем выбора значения объекта вполне могли быть получены новые факты применением многочисленных продукций. Возникающая при этом потребность в отмене выбранного значения и перехода на шаг назад в диалоге достигается построением дополнительных структур, содержащих хронологию логического вывода.

Модуль интерфейса класса *LPStructure* обеспечивает расширенные возможности пакета в плане разработки и исследования БЗ. Эти возможности включают лишь при обнаружении на компьютере доступного файла *LPStructure.dll*. В этом случае данная библиотека динамически загружается и модуль настраивает связи с ее функциями.

При наличии *LPStructure.dll* по окончании компиляции БЗ для правил в памяти создаются по два соответствующих двоичных вектора (предпосылка и заключение). Эти векторы (точнее, их адреса) в дальнейшем могут быть использованы при формировании *LP*-структуры. Размерность векторов вычисляется на основе результатов компиляции. Точками соответствующей решетки будут пары "*объект* = *значение*". Для создания *LP*-структуры модуль формирует набор пар логического отношения [4]. Далее можно использовать все возможности класса *LPStructure*, а именно: выявлять избыточные правила; исследовать образы и прообразы заданных подмножеств фактов; осуществлять обратный вывод на основе решения логических уравнений [5]. В дополнение к перечисленным имеется также возможность проверки множества фактов на непротиворечивость, т. е. отсутствие в нем одновременно нескольких значений многозначного объекта.

Часть интерфейсного модуля поддерживает реализацию *LP*-вывода. Этот обратный вывод представляет собой альтернативу методу, который реализуется описанной выше машиной вывода. Он осуществляется путем построения минимальных начальных прообразов в *LP*-структуре [5] для разрешенных значений объекта экспертизы с последующим их исследованием. Предпочтение отдается решениям, которые требуют меньшего количества вопросов пользователю.

Таким образом, пакет *LPExpert* представляет собой полноценную интегрированную систему логического программирования с новыми высокопроизводительными средствами автоматизации и отладки.

Список литературы

1. **Sowyer B., Foster D.** Programming Expert Systems in Pascal. John Wiley & Sons, Inc., 1986.
2. **Giraud-Carrier C. G., Martinez T. R.** An Integrated Framework for Learning and Reasoning // Journal of Artificial Intelligence Research. 1995. V. 3. P. 147—185.
3. **Ашиняц Р. А., Засыпкин А. В.** Проект гибридной экспертной системы ОХРАНА // Тез. докл. 2-й Всесоюзной школы-семинара "Экспертные системы и Пролог в учебном процессе". — Йошкар-Ола, 1990. — С. 4—9.
4. **Махортов С. Д.** Логические отношения на решетках // Вестник ВГУ. Серия. "Физика, математика". — Воронеж, 2003. № 2. — С. 203—209.
5. **Махортов С. Д.** Логические уравнения на решетках // Вестник ВГУ. Серия. "Физика, математика". — Воронеж, 2004. № 2. — С. 170—178.

А. А. Кирьянов*, инженер,
В. Ю. Сироткин**, программист,
Ю. В. Сироткин**, вед. инженер,
А. А. Смирнова*, инженер,

*"МАТИ"-Российский государственный
 технологический университет
 им. К. Э. Циолковского
 **ООО "Связь-Строй",
 г. Москва, e-mail:5079311@mail.ru

Программный комплекс для автоматизированной системы мониторинга экологического состояния на предприятии

Приведены результаты разработки программного комплекса, предназначенного для автоматизированной системы мониторинга экологического состояния на предприятии. Комплекс позволяет автоматизировать обработку информации с датчиков состояния среды, интегрировать ее в единую базу данных и уведомлять оператора как о превышении допустимых концентраций вредных веществ, так и о тенденциях изменения экологической обстановки.

Ключевые слова: программный комплекс, экологический мониторинг предприятия.

Эффективность мониторинга экологической ситуации на предприятии может совершенствоваться за счет автоматизация обработки значительных объемов информации, поступающей от значительного числа различных датчиков и характеризующей наличие тех или иных примесей в воздухе. В настоящее время на этой основе созданы программные комплексы [1—3], позволяющие автоматизировать обработку информации в системах экологического мониторинга. При этом основное внимание уделяется контролю над состоянием окружающей среды в конкретном регионе с привязкой к местности.

Целью данной работы являлась разработка программного комплекса, предназначенного для обработки, передачи и хранения информации об экологическом состоянии в рабочих помещениях предприятий.

Программный комплекс написан на языке программирования C++ и функционирует под управлением операционной системы с открытым кодом Linux Red Hat Enterprise Server 5.2 с

поддержкой графической системы GNOME. Функциональная структурная схема разработанного комплекса представлена на рисунке.

Структура программного комплекса

Все модули программного комплекса можно разделить на три рабочих группы:

- система для сбора данных экологического состояния окружающей среды;
- система управления базой данных;
- автоматизированное рабочее место оператора (АРМ).

Система для сбора данных экологического состояния окружающей среды представляет собой комплекс технических и программных средств, предназначенных для сбора и первичной обработки данных об экологических параметрах контролируемых компонентов природной среды. Контроль за компонентами среды осуществляется как путем непрерывных измерений с помощью устройств слежения за концентрациями вредных веществ (датчиков), так и посредством отбора проб для проведения последующих анализов в аналитической лаборатории. В систему входит модуль преобразования машинограмм внешних устройств к формату разрабатываемого программного комплекса, куда передается информация с датчиков угла, газа, давления и т. п. Данный модуль работает в серверном режиме, независимо от работы и задач, выполняемых на АРМ.

Входящая в состав разработанного программного комплекса система управления базами данных позволяет организовывать накопление, обработку, хранение и защиту полученной в системе сбора информации, проводит операции по извлечению, обновлению, добавлению и удалению, а также поиск информации по запросам пользователей. Система предусматривает хранение полученных данных за последние пять лет.

АРМ взаимодействует с модулями, связанными с работой интерфейса, модулями статистики, анализа и прогнозирования и модулем транспортного характера.



Структурная схема программного комплекса

К модулям, связанным с работой интерфейса в данном программном комплексе, относятся:

- система определения прав доступа, которая обеспечивает возможность управления и получения статистических данных в общих сетях предприятия;
- модуль модельного анализа виртуальной экологической карты;
- модуль статистической обработки исходных данных;
- модуль принятия решений;
- модуль интерфейса редактора математической модели.

В модуле модельного анализа виртуальной экологической карты осуществляется прогнозирование возникновения аварий и выявление точек повышенной опасности. Данный программный компонент значительно ускоряет и облегчает процесс обнаружения очагов загрязнения. В модуле статистической обработки исходных данных происходит анализ тенденций изменения концентраций вредных веществ на территории предприятия. При этом происходит последовательный перебор всех объемов сетки, к каждому из которых применяются критерии локализации частицы. Анализ данных, полученных в модуле статистической обработки, происходит в модуле принятия решения. Полученные значения концентрации вредных веществ сравниваются с допустимыми концентрациями, а также с концентрациями, полученными на предыдущих этапах.

К модулям и алгоритмам статистики, анализа и прогнозирования относятся:

- модуль формирования банка данных с виртуальными тест-объектами, где происходит эмуляция реальной ситуации за счет генерации машинограмм, которые необходимы для отладки и проверки работоспособности программного комплекса;
- алгоритм прогнозирования возможной аварии;

- алгоритм анализа аварии для локализации источников загрязнения.

Модуль взаимодействия с контрольно-экологической системой (модуль транспортного характера) является связующим звеном между сохраненными и преобразованными исходными данными и всеми рабочими модулями, обеспечивающим обработку и анализ этих данных.

При ухудшении экологической обстановки система подает звуковой сигнал, и на экране монитора компьютера появляется информация, указывающая на зоны, в которых отмечаются тенденции к повышению концентраций вредных веществ.

Вывод

Разработанный программный комплекс позволяет собрать всю информацию с различных датчиков в единую базу данных и в короткие сроки создавать виртуальную карту, достаточно полно отражающую картину экологического состояния предприятия.

Программный продукт был разработан на предприятии ООО "Связь-Строй".

По вопросам использования обращаться по телефону: (495) 507-93-11.

Список литературы

1. Ephraim Goldenberg, Shaul Serero, David Cohen, Yechiel Spector, Esther Jacobson. Method and system for detecting gases or vapors in a monitored area. Патент США № 6061141, 2000.
2. Алатырцев А. Б., Безрук Д. Б., Князева В. В., Марасанов А. М., Пирумов У. Г., Рыбачук О. С., Юров Н. Н., Шейпак С. А. Программный комплекс "Гарант-Универсал" // Экология производства. № 1. 2007. С. 50—55.
3. Банин В. Н., Герасимов И. А., Губяк В. Е. Система экологического контроля окружающей среды. Патент РФ, № 2078357, 2005.

ДИСКУССИОННЫЙ КЛУБ

УДК 519.7

В. Б. Вяткин, канд. техн. наук, г. Екатеринбург, e-mail: vbv@yandex.ru

Синергетический подход к определению количества информации

Излагается новый подход к определению количества информации, в котором за информацию принимаются сведения о конечном множестве как едином целом, а мерой информации является средняя длина интегративного кода элементов.

Ключевые слова: негэнтропия, количество информации, отражение, конечное множество, элемент, признак, интегративный код.

Введение

Исследуя ту или иную дискретную систему, мы, как правило, по отличительным признакам выделяем в ее составе совокупность частей, представляющих собой конечные множества элементов. В том случае, когда какие-либо части системы, выделенные в плоскостях различных признаков, имеют непосредственную взаимосвязь

друг с другом, наблюдается пересечение соответствующих множеств элементов. При этом очевидно, что любые два пересекающихся множества отражают (воспроизводят) друг о друге, как о целостном образовании, определенную информацию, количественная оценка которой представляет практический интерес. С гносеологической точки зрения процесс получения этой информации

Рисунок 1:

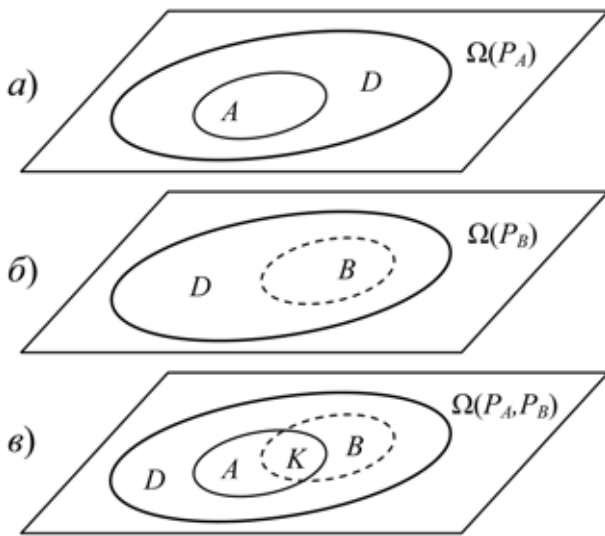


Рис. 1. Система D и множества A, B, K в плоскостях признаков P_A и P_B

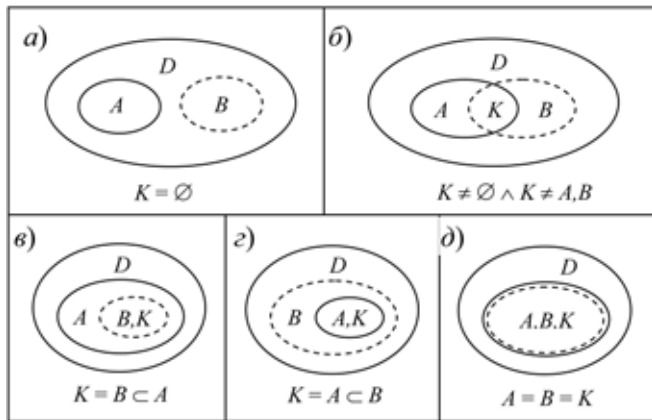


Рис. 2. Модели взаимосвязи множеств A и B в составе системы D : a — модель отсутствия взаимосвязи; $б, в, г$ — модель частичной (статистической) взаимосвязи; $д$ — модель полной (взаимно однозначной) взаимосвязи

познающим субъектом (аналитической системой) состоит из трех этапов, и на примере произвольной системы D и отличительных признаков P_A и P_B выглядит следующим образом.

На первом этапе система D рассматривается в плоскости $\Omega(P_A)$ признака P_A (рис. 1, a) и те элементы $d \in D$, у которых наблюдается признак P_A , выделяются в виде множества A . На втором этапе идет рассмотрение системы в плоскости $\Omega(P_B)$ признака P_B (рис. 1, $б$) и аналогично выделяется множество B .

После завершения операций первых двух этапов познающий субъект находится в состоянии неопределенности относительно существования непосредственной взаимосвязи между множествами A и B . Эта неопределенность снимается на третьем этапе после рассмотрения выделенных множеств A и B в совмещенной плос-

кости $\Omega(P_A, P_B)$ признаков P_A и P_B (рис. 1, $в$). Если при этом выявляется третье (связующее) множество K такое, что $K = A \cap B$, $K \neq \emptyset$, то познающий субъект снимает (ликвидирует) свою неопределенность относительно непосредственной взаимосвязи множеств A и B и получает информацию (I_{AB}), которую эти множества отражают друг о друге. В противном случае, когда $K = \emptyset$, отмеченная неопределенность также снимается, но при этом делается вывод, что между множествами A и B существует только косвенная взаимосвязь, заключающаяся в том, что как A , так и B принадлежат одной и той же системе D .

В настоящее время синонимом неопределенности чего-либо, а также отсутствия или недостатка знаний (информации) о чем-либо принято считать энтропию [1, 2, 3]. Поэтому информацию I_{AB} , для ее отличия от других видов информации, будем называть негэнтропией отражения (принимая при этом во внимание также негэнтропийный принцип Бриллюэна, согласно которому "информация представляет собой отрицательный вклад в энтропию" [4, с. 34]). То есть негэнтропия отражения представляет собой информацию I_{AB} , которую отражают друг о друге два пересекающихся конечных множества A и B . Желая определить, чему равна негэнтропия отражения I_{AB} , поставим задачу по ее количественному определению следующим образом.

Пусть в составе некоторой системы $D = \{d\}$ (рис. 2) по отличительным признакам P_A и P_B выделены три конечных множества $A = \{a|P_A(a)\} = \{d|P_A(d)\}$, $B = \{b|P_B(b)\} = \{d|P_B(d)\}$ и $K = A \cap B$, $K \neq \emptyset$. Число элементов в составе каждого из множеств равно M_A, M_B, M_K соответственно. Требуется определить, чему равна негэнтропия отражения I_{AB} , т. е. количество информации, которую отражают друг о друге конечные множества A и B .

Актуальность количественной оценки негэнтропии отражения во многих предметных областях не вызывает сомнений (например, при решении задач, связанных с оценкой информативности признаков). Вместе с тем установлено [5], что известные подходы к определению количества информации [6] не позволяют решить поставленную задачу. Более того, широко известная информационно-энтропийная мера Шеннона [7] при наиболее полной взаимосвязи конечных множеств A и B (рис. 2, $д$), когда они должны отражать друг о друге максимальное количество информации, приводит к нонсенсу, показывая, что $I_{AB} = 0$. Это объясняется тем, что математические основы теории информации, начиная с работ Хартли [8], традиционно разрабатывались под эгидой того, что информация неразрывно связана с управлением и представляет собой снимаемую неопределенность выбора одной из множества возможностей. Другой же вид информации, объективно существующий в природе независимо от управления и не связанный с выбором [9], при этом остался в тени. К этому виду информации, по всей видимости, относится и негэнтропия отражения, для количественной оценки которой автором статьи разработан новый — синергетический — подход к определению количества информации. Ниже дается изложение этого подхода.

Самоотражение конечных множеств

Анализируя модели взаимосвязи конечных множеств A и B (рис. 2), можно утверждать, что при постоянстве M_A и M_B негэнтропия отражения I_{AB} увеличивается по

мере роста M_K и является максимальной, когда $A = B = K$. В этом случае отражение множеств A и B друг через друга не отличается от их самоотражения, т. е. отражения через самих себя. Соответственно, негэнтропия I_{AB} при $A = B = K$ равна самоотражаемой информации, которую каждое из множеств отражает о самом себе как едином целом. Это говорит о том, что негэнтропия отражения и информация, самоотражаемая конечными множествами, имеют одну и ту же природу. Поэтому, прежде чем непосредственно решать поставленную задачу по оценке негэнтропии отражения, рассмотрим информационные аспекты самоотражения конечных множеств.

Будем исходить из общеупотребительного и наиболее простого определения информации как сведений о чем-либо и примем в наших исследованиях за информацию сведения о конечном множестве как едином целом. При этом, используя словосочетание "единое целое", мы имеем в виду, что, во-первых, конечное множество в контексте его отражения является неделимым, а во-вторых, элементы множества в своей совокупности представляют не механическое собрание предметов, существующих независимо друг от друга, а целостное образование, в составе которого элементы обладают интегративными характеристиками, не присущими им в их разобленном виде. Короче говоря, показателем конечного множества как единого целого являются интегративные характеристики его элементов. Соответственно, наличие у этих характеристик какого-либо числового параметра, зависящего от общего числа элементов, может служить основой для определения количества информации, самоотражаемой конечным множеством. Определим это количество информации, для чего примем следующий аксиоматический базис.

1. Информация представляет собой сведения о конечном множестве элементов как едином целом.

2. Количество информации I_A , самоотражаемой конечным множеством A , является монотонно возрастающей функцией от общего числа его элементов M_A и, соответственно, для любых двух конечных множеств A и B с числом элементов M_A и $M_B = M_A + 1$ имеет место неравенство

$$I_B > I_A. \quad (1)$$

3. Показателем конечного множества A как единого целого является интегративный код его элементов, представляющий собой индивидуальную для каждого элемента последовательность символов какого-либо алфавита, число которых L_A (длина кода) является функцией от общего числа элементов M_A в составе множества.

Рассмотрим процесс увеличения числа элементов M_A , представив его в виде роста ориентированного дерева, совокупность висячих вершин которого взаимно однозначно соответствует множеству элементов $a \in A$, а максимальное число дуг, выходящих из одной вершины, равно числу символов n алфавита, выбранного для составления интегративных кодов. При этом каждой из смежных дуг в алфавитном порядке ставится в соответствие свой символ и, как следствие, в качестве индивидуального интегративного кода какого-либо элемента выступает последовательность символов, находящихся на пути движения из начальной вершины дерева в соответствующую данному элементу висячую вершину.

Модель такого дерева, которое будем называть деревом кодов, при $n = 2$ и использовании в качестве алфа-

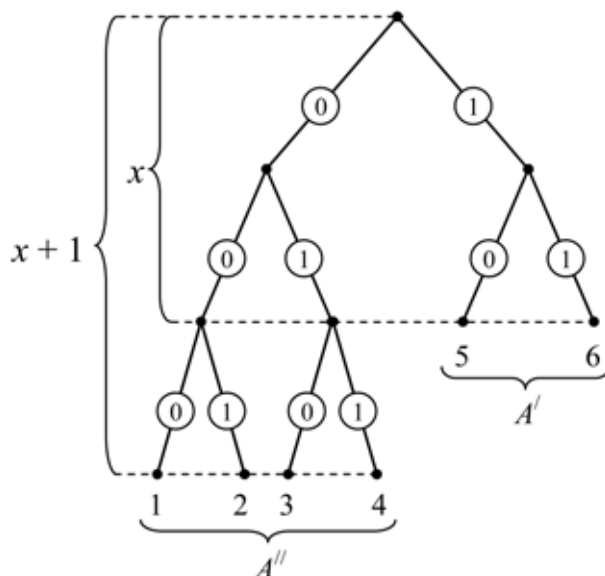


Рис. 3. Модель дерева кодов при $n = 2$ и $M_A = 6$

вита упорядоченной пары символов $\langle 0, 1 \rangle$ приведена на рис. 3. Из рисунка видно, что в общем случае множество A по длине интегративных кодов его элементов разбивается на два подмножества A' и A'' , таких что $L_{A'} = x$ и $L_{A''} = x + 1$, где $x = \lfloor \log_2 M_A \rfloor$ — целочисленная часть $\log_2 M_A$. То есть L_A не является однозначной функцией от M_A . Поэтому будем рассматривать среднюю длину (\bar{L}_A) интегративных кодов

$$\bar{L}_A = \frac{xM_{A'} + (x+1)M_{A''}}{M_A} \quad (2)$$

и начнем с алфавита с минимальным числом символов ($n = 2$).

Из рис. 3 видно, что при $n = 2$ возрастание M_A на единицу обуславливает уменьшение на единицу числа элементов с длиной кода x и увеличение числа элементов с длиной кода $x + 1$ на два элемента, т. е.

$$M_A + 1|_{n=2} \Rightarrow (M_{A'} - 1) \wedge (M_{A''} + 2). \quad (3)$$

Учитывая (3), для определения $M_{A'}$ и $M_{A''}$ составим систему уравнений

$$\begin{cases} M_{A'} + M_{A''} = M_A; \\ 2M_{A'} + M_{A''} = 2^{x+1}, \end{cases}$$

решая которую, получаем:

$$\begin{cases} M_{A'} = 2^{x+1} - M_A; \\ M_{A''} = 2(M_A - 2^x). \end{cases} \quad (4)$$

Подставляя значения (4) в выражение (2) и проводя несложные преобразования, приходим к следующей формуле средней длины интегративных кодов при $n = 2$:

$$\bar{L}_A|_{n=2} = x + 2 - \frac{2^{x+1}}{M_A}. \quad (5)$$

Полученное выражение (5) удовлетворяет принятым аксиомам и, соответственно, может служить мерой количества информации I_A , самоотражаемой конечным множеством A .

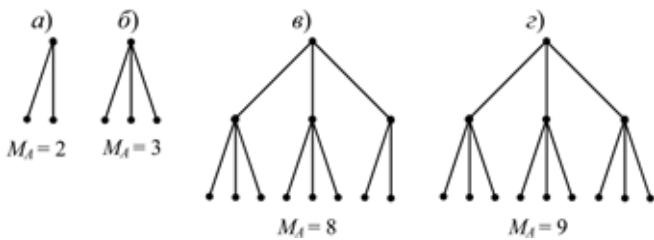


Рис. 4. Модели дерева кодов при $n = 3$

Рассмотрим теперь деревья кодов при $n > 2$. На рис. 4 представлены такие деревья, когда $n = 3$ и $M_A = 2, 3, 8, 9$.

Из рисунка видно, что при наполнении выходящими дугами начальной вершины дерева (рис. 4, а, б) и последней из висячих вершин (рис. 4, в, г) средняя длина кодов \bar{L}_A не изменяется, т. е.

$$n = 3 \Rightarrow \begin{cases} \bar{L}_A|_{M_A=2} = \bar{L}_A|_{M_A=3} = 1; \\ \bar{L}_A|_{M_A=8} = \bar{L}_A|_{M_A=9} = 2; \\ \dots\dots\dots \\ \bar{L}_A|_{M_A=3^y-1} = \bar{L}_A|_{M_A=3^y} = y, \end{cases}$$

где $y = 1, 2, \dots$

Увеличивая n , приходим к общему выражению случаев постоянства значений \bar{L}_A при наполнении выходящими дугами последних из висячих вершин:

$$n > 2 \Rightarrow \bar{L}_A|_{M_A=n^y-n+2} = \bar{L}_A|_{M_A=n^y-n+3} = \dots = \bar{L}_A|_{M_A=n^y} = y. \quad (6)$$

Из выражения (6) следует, что \bar{L}_A при $n > 2$ и $M_A \geq n^y$ не менее чем в $(n-2)u$ случаев противоречит аксиоме (1) монотонного возрастания информации I_A . Это позволяет сделать принципиально важный вывод: средняя длина интегративного кода элементов может выступать в качестве меры количества информации, самоотражаемой конечным множеством, только тогда, когда интегративные коды составлены с помощью двоичного алфавита.

Таким образом, мы пришли к тому, что

$$I_A = \bar{L}_A|_n = 2 \quad (7)$$

и, соответственно, все излагаемое ниже будет относиться к $n = 2$.

Проводя анализ формулы (5), нетрудно видеть, что если $M_A = 2^x$, то $\bar{L}_A = \log_2 M_A$. В тех же случаях, когда $2^x < M_A < 2^{x+1}$, наблюдается некоторое превышение \bar{L}_A над $\log_2 M_A$. Определим максимальную величину этого превышения (Ψ) как точную верхнюю грань отклонения \bar{L}_A от $\log_2 M_A$:

$$\Psi = \sup_{M_A \in [1, \infty)} (\bar{L}_A - \log_2 M_A).$$

Применяя необходимое условие экстремума функции и полагая, что $M_A \in (2^x, 2^{x+1})$, $x = \text{const}$, в соответствии с выражением (5) приходим к уравнению

$$\left(x + 2 - \frac{2^{x+1}}{M_A} - \log_2 M_A\right)'_{M_A} = 0,$$

которое после дифференцирования по M_A приобретает вид

$$\frac{2^{x+1}}{M_A^2} - \frac{1}{M_A \ln 2} = 0$$

и после несложных преобразований имеет своим решением

$$M_A = 2^{x+1} \ln 2. \quad (8)$$

Подставляя значение M_A из (8) в разность $(\bar{L}_A - \log_2 M_A)$ и представляя при этом \bar{L}_A в развернутом виде, имеем:

$$\begin{aligned} \bar{L}_A - \log_2 M_A &= x + 2 - \frac{2^{x+1}}{2^{x+1} \ln 2} - \log_2 2^{x+1} - \\ &- \log_2(\ln 2) = x + 2 - \frac{1}{\ln 2} - x - 1 - \log_2(\ln 2) = \\ &= 1 - \frac{1}{\ln 2} - \log_2(\ln 2). \end{aligned}$$

Так как в соответствии со свойствами логарифмов $\log_2(\ln 2) = \frac{\ln(\ln 2)}{\ln 2}$, то из последнего выражения окончательно получаем, что отклонение значений \bar{L}_A от $\log_2 M_A$ ограничено постоянной величиной

$$\Psi = 1 - \frac{1 + \ln(\ln 2)}{\ln 2} = 0,0860713\dots, \quad (9)$$

что наглядно иллюстрирует рис. 5.

Значение полученной постоянной (9) позволяет сделать приближение:

$$2^x < M_A < 2^{x+1} \Rightarrow \bar{L}_A \approx \log_2 M_A. \quad (10)$$

Функция $f = \log_2 M_A$ является монотонно возрастающей и ее значения удовлетворяют информационной аксиоме (1). Поэтому, основываясь на выражении (7) и принимая во внимание более простой вид \bar{L}_A в выражении (10) по сравнению с выражением (5), окончательно примем меру информации I_A в следующем виде:

$$I_A = \log_2 M_A. \quad (11)$$

Полученная формула количества информации, самоотражаемой конечным множеством (11), математически подобна информационной мере Хартли [8], взятой при единичном выборе и двоичном основании логарифма, но принципиально отличается от нее тем, что число символов используемого алфавита в мере Хартли является аргументом логарифма, а в формуле (11) — его основа-

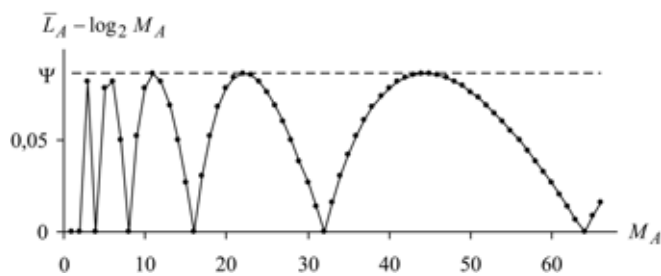


Рис. 5. График разности $\bar{L}_A - \log_2 M_A$

нием. Кроме того, основание логарифма в мере Хартли может быть любым, а в мере (11) основание больше двух не допускается. В связи с этим уместно привести высказывание академика А. Н. Колмогорова о математическом подобии различных информационно-энтропийных функций: "Такие математические аналогии следует всегда подчеркивать, так как сосредоточение на них внимания содействует прогрессу науки" [10, с. 39].

Подводя итог сделанному в настоящем разделе, можно констатировать, что нами получен новый подход к определению количества информации, основанный на совместном и одновременном учете всех элементов конечного множества, без выделения какого-либо из них в качестве случайного события, результата испытания и т. п. То есть мы ушли от традиционного увязывания количества информации с результатами выбора одной из множества различных возможностей. Чтобы отличать этот подход от других подходов к количественному определению информации, будем называть его синергетическим подходом. Такое название обусловлено тем, что количество информации, самоотражаемой конечным множеством, является результатом совместного отражающего действия всей совокупности его элементов, а термин "синергетика" буквально означает "совместный, согласованно действующий".

Негэнтропия отражения конечных множеств

Определив информацию I_A как величину \bar{L}_A , мы тем самым предвосхитили оценку негэнтропии отражения I_{AB} как результат воспроизведения средней длины интегративного кода элементов одного конечного множества через пересекающееся с ним другое конечное множество. Непосредственно оценивая теперь величину I_{AB} , будем исходить из того, что множества A , B , K в своей совокупности образуют простейшую систему информационной связи, в которой отражаемое (A) и отражающее (B) множества являются, соответственно, источником и приемником информации, а связующее множество $K = A \cap B$ выступает в качестве передающей среды или канала связи. Процесс передачи информации I_A по такой системе связи соответствует процессу отражения множества A через непосредственно взаимосвязанное с ним множество B . Рассмотрим этот процесс, учитывая, что вне связи с управлением взаимодействующие объекты участвуют во взаимном отражении друг друга всей совокупностью своих элементов [9].

Интегративный код любого элемента $a \in A$ представляет собой определенное сообщение о конечном множестве A как о целостном образовании, а общее число таких сообщений равно M_A . Соответственно, объем информации V_A , направляемый отражаемым множеством A в связующее множество K , равен

$$V_A = I_A M_A.$$

Пропускная способность множества K в силу того, что $K \subset A$, не может превышать V_A и ограничена объемом информации

$$V_K = I_K M_K.$$

Вследствие этого информация I_A при поступлении в множество K уменьшается до величины

$$I_{A \rightarrow K} = \frac{V_K}{M_A} = I_K \frac{M_K}{M_A}.$$

В отражающее множество B из связующего множества K поступает M_K сообщений, которые несут об отражаемом множестве A объем информации

$$V_{A \rightarrow B} = I_{A \rightarrow K} M_K = I_K \frac{M_K^2}{M_A},$$

и, соответственно, информация I_A воспроизводится множеством B в виде информации

$$I_{A \rightarrow B} = \frac{V_{A \rightarrow B}}{M_B} = I_K \frac{M_K^2}{M_A M_B}. \quad (12)$$

Если теперь рассмотреть обратный процесс отражения множества B через множество A , то мы получим такой же результат (12), т. е. $I_{A \rightarrow B} = I_{B \rightarrow A} = I_{AB}$.

Таким образом, мы пришли к тому, что формула негэнтропии отражения I_{AB} имеет вид

$$I_{AB} = \frac{M_K^2}{M_A M_B} \log_2 M_K. \quad (13)$$

Из выражения (13) следует, что I_{AB} является частью средней длины интегративного кода элементов связующего множества K . Так как $K \subset A$, B , то можно также утверждать, что негэнтропия отражения I_{AB} в количественном отношении является мерой воспроизведения средней длины интегративного кода элементов каждого из пересекающихся множеств A и B .

Заключение

Проведение изложенных информационно-теоретических исследований было обусловлено выявлением фактов неустойчивости и противоречивости результатов прогнозно-геологических исследований при оценке информативности признаков с помощью теории информации Шеннона [11]. Эти неустойчивость и противоречивость устраняются, если признаки оценивать с помощью негэнтропии отражения конечных множеств [5], что может представлять интерес для специалистов в области распознавания образов при оценке информативности признаков эталонных объектов распознавания.

При выводе формулы информации, самоотражаемой конечным множеством, и оценке негэнтропии отражения мы оперировали единичными значениями признаков, рассматривая их как некоторые отличительные особенности элементов. Вместе с тем, признаки описания элементов какой-либо системы в общем случае имеют различные значения, число которых может доходить до общего числа элементов в составе системы. Поэтому дальнейшее развитие синергетического подхода к определению количества информации, по нашему мнению, будет происходить в плоскости анализа отражения дискретных систем через совокупности своих частей, выделенных по различным значениям признаков описания их элементов. При этом можно ожидать выявления определенной взаимосвязи с вероятностной теорией информации, поскольку формула самоотражаемой информации математически подобна мере Хартли, а последняя является частным случаем информационно-энтропийной меры Шеннона, когда все возможности имеют одинаковую вероятность.

Список литературы

1. Шамбадаль П. Развитие и приложение понятия энтропии. М.: Наука, 1967. 280 с.
2. Волькенштейн М. В. Энтропия и информация. М.: Наука, 1986. 192 с.
3. Седов Е. А. Одна формула и весь мир. Книга об энтропии. М.: Знание, 1982. 176 с.
4. Бриллюэн Л. Научная неопределенность и информация. М.: Мир, 1966. 272 с.
5. Вяткин В. Б. Математические модели информационной оценки признаков рудных объектов: Автореф. дис. ... канд. техн. наук: 05.13.18: Екатеринбург, 2004. 27 с. Режим доступа: http://orel3.rsl.ru/dissert/EBD_1639A_vjatkinVB.pdf

6. Колмогоров А. Н. Три подхода к определению понятия "количество информации" // Проблемы передачи информации. 1965. Т. 1. № 1. С. 3—11.
7. Шеннон К. Работы по теории информации и кибернетике. М.: Изд-во иностр. лит., 1963. 830 с.
8. Хартли Р. В. Л. Передача информации // Сб.: Теория информации и ее приложения. М.: Физматгиз, 1959. С. 5—35.
9. Урсул А. Д. Проблема информации в современной науке. М.: Наука, 1975. 288 с.
10. Колмогоров А. Н. Теория информации и теория алгоритмов. М.: Наука, 1987. 304 с.
11. Вяткин В. Б. К вопросу информационной оценки признаков при прогнозно-геологических исследованиях // Известия Уральского горного института. Сер.: Геология и геофизика. 1993. Вып. 2. С. 21—28.

Указатель статей, опубликованных в журнале "Информационные технологии" в 2009 году

ВЫЧИСЛИТЕЛЬНЫЕ УСТРОЙСТВА, СИСТЕМЫ И СЕТИ (ЭВМ И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ)

- Аверичева Д. Л., Семенов А. С., Фролов А. С. Поиск вширь в графе на суперкомпьютере с мультитредово-поточковой архитектурой. № 7.
- Аль-Аммори Али. Оптимизация параллельного информационного резервирования методом вложенных модулей. № 2.
- Аристархов В. Ю. Алгоритм приема сигнала в целом для высокоскоростных беспроводных сетей. № 4.
- Асратян Р. Э. Метод организации Web-сервисных взаимодействий между удаленными частными сетями. № 6.
- Барский А. Б. Применение логических нейронных сетей для выбора оптимальной стратегии обслуживания потока запросов в системе GRID-вычислений. № 1.
- Бобков С. Г. Методы повышения эффективности контроллера коммутатора Ethernet 10/100 Мбит/с. № 2.
- Бобков С. Г. Высокоэффективный адаптер коммуникационной среды многопроцессорной ЭВМ. № 4.
- Богатырев В. А., Богатырев С. В. Объединение резервированных серверов в кластеры высоконадежной компьютерной системы. № 6.
- Богатырев В. А., Богатырев С. В. Надежность резервированной двухуровневой компьютерной системы при ограниченном времени обслуживания запросов. № 7.
- Брындин Е. Г. Теоретические аспекты технологии непрерывной обработки на виртуальной памяти. № 9.
- Вашкевич Н. П., Бикташев Р. А. Формализация алгоритма синхронизации процессов при диспетчеризации задач в многопроцессорных системах с использованием механизма "рандеву". № 12.
- Грушин А. И., Ремизов М. Л., Ростовцев А. В., Николаев Д. Д., Чинь Куанг Киен. Высокопроизводительное устройство для обработки радиолокационной информации. № 7.
- Ким А. К., Фельдман В. М. Вычислительная система реального времени для реализации программ управления сложными объектами. № 2.
- Коваленко О. Н. Влияние метода обработки очередей на качество обслуживания трафика мультисервисной сети. № 12.

- Кристовский Г. В., Погребной Ю. Л., Соин С. А. Разработка и исследование маломощного быстродействующего трехпортового регистрового файла. № 2.
- Мамченко А. Е. Образовательно-методологический аспект "принстонской" и "гарвардской" архитектур процессоров вычислительных систем. № 7.
- Путря Ф. М. Архитектурные особенности процессоров с большим числом вычислительных ядер. № 4.
- Размахнин С. А., Куприянов А. И. Алгоритм разработки систем оперативно-розыскных мероприятий для сервисов, построенных на базе технологий мобильной связи. № 4.
- Сериков Д. А. К математическому моделированию процесса диспетчеризации задач в распределенной вычислительной среде. № 12.
- Сигарев А. А. Методология упразднения межпроцессорного обмена в MVC со множественным потоком команд. № 7.
- Силина А. Ю., Васильева В. Д., Дербишер В. Е., Гермашев И. В. Систематизация наукометрических показателей эффективности научной деятельности. № 6.
- Шкунов В. И. О методологии оценки и сравнения характеристик различных протоколов беспроводных сетей с переменной топологией. № 1.

СЕТИ И СИСТЕМЫ СВЯЗИ

- Гечис А. К., Соколова О. Д., Соколов Н. А. Входящий поток заявок для голосового трафика в сетях следующего поколения. № 4.
- Мочалов В. А. Алгоритмы размещения транзитных узлов в сенсорной сети. № 10.
- Наумова В. В., Сорокин А. А., Горячев И. Н. Видеоконференцсвязь — мультимедийный сервис корпоративной сети Дальневосточного отделения РАН. № 4.
- Огнев В. А., Иванов С. Р. Методы повышения помехоустойчивости аппаратуры потребителей спутниковых навигационных систем. № 4.
- Огнев В. А., Иванов С. Р. Математическая модель приема-вычисления систем ГЛОНАСС/GPS. № 10.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Ахремчик О. Л. Информационная база для автоматизированного проектирования схем систем управления технологическими объектами. № 8.
- Гаврилов С. В., Каграманян Э. Р., Ходош Л. С. Тенденции развития моделей библиотечных элементов для статического временного анализа цифровых СБИС. № 3.

Ильин В. Н., Гришин Р. А. Методика оценки основных параметров цифровых устройств на ПЛИС на ранних этапах проектирования с использованием двухуровневого макромоделирования. № 7.

Мосин С. Г. Современные тенденции и технологии проектирования интегральных схем. № 1.

Немтинов В. А., Егоров С. Я., Пахомов П. И. Применение теории нечетких множеств и экспертных систем при автоматизированном выборе элемента технической системы. № 10.

Петунии А. А. Автоматизация нерегулярного фигурного раскроя: основные подходы к разработке систем проектирования раскроя и управляющих программ для машин резки листовых материалов. № 10.

Стемпковский А. Л., Глебов А. Л., Гаврилов С. В., Гудкова О. Н. Вероятности напряженного состояния транзисторов для временного анализа с учетом электротемпературной нестабильности. № 7.

Талицкий Е. Н. Алгоритм проектирования виброзащиты электронной аппаратуры. № 1.

Шмырев Н. В. Средства моделирования и контроля распределенных систем реального времени. № 3.

Шумарин С. В., Талицкий Е. Н. Оптимизация виброзащиты электронной аппаратуры методом частотной отстройки. № 8.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

Архипова О. Е. Концепция региональной эколого-информационной системы мониторинга. № 5.

Бушмелева К. И., Плюснин И. И. Автоматизированная геоинформационная система мониторинга технического состояния магистральных газопроводов. № 5.

Кобзаренко Д. Н., Камилова А. М., Гаджимурадов Р. Н. Концепция построения системы трехмерного геоинформационного моделирования. № 11.

Черняев А. В., Павлов А. А. Моделирование процессов осаждения нефтяных загрязнений на береговую поверхность малых рек. № 11.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

Авдошин С. М., Савельева А. А. О новом подходе к проблеме анализа эффективности криптосистем. № 8.

Андреев О. О. Интеграция моделей логического разграничения доступа, описанных на специализированном языке. № 12.

Бочков М. В., Шкадов А. А. Формальная модель состояний системы защиты компьютерной сети при использовании политик информационной безопасности. № 1.

Васенин В. А. Критическая энергетическая инфраструктура: кибертеррористическая угроза. № 9.

Васенин В. А. Критическая энергетическая инфраструктура: средства противодействия кибертерроризму. № 10.

Жуков О. Д. Модулярные числовые системы в криптографии. № 2.

Зегжда П. Д., Зегжда Д. П., Калинин М. О. Реализация логического подхода к оценке безопасности состояния ОС семейства MS Windows в системе анализа настроек безопасности "Декарт". № 2.

Котенко И. В., Воронцов В. В., Чечулин А. А., Уланов А. В. Проактивные механизмы защиты от сетевых червей: подход, реализация и результаты экспериментов. № 1.

Котенко И. В., Уланов А. В. Многоагентное моделирование механизмов защиты от распределенных компьютерных атак. № 2.

Молдовян Н. А., Молдовяну П. А. Конечные группы векторов, содержащие подгруппы простого порядка большого размера. № 1.

Найденко В. Г. О неявной аутентификации пользователей компьютерных сетей. № 6.

Перегида А. И., Тимашов Д. А. Математическая модель надежности систем защиты информации. № 8.

Ремизов А. В., Филиппов М. В., Чичварин Н. В. Методы защиты информации в звуковых файлах. № 10.

Суханов А. В. Подход к построению защищенных информационных систем. № 6.

Ушмаев О. С. Проблемы применения биометрической идентификации в комплексной защите информации. № 12.

Шапченко К. А. Способ проверки свойств безопасности в моделях логического разграничения доступа с древовидной иерархией объектов доступа. № 10.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

Алыгулиев Р. М. Реферирование документов через ранжирование предложений. № 10.

Арзамасцев А. А., Зенкова Н. А., Неудахин А. В. Технология построения медицинской экспертной системы на основе аппарата искусственных нейронных сетей. № 8.

Артемьева И. Л. Сложно структурированные предметные области. Построение многоуровневых онтологий. № 1.

Брындин Е. Г. Теоретические основы коммуникативно-ассоциативной имитации символически-языкового мышления. № 2.

Васенин В. А., Афонин С. А., Козицын А. С. Автоматизированный анализ текстовой информации. № 7.

Гольдштейн С. Л., Кудрявцев А. Г. Проблематика создания системного интеллектуального подсказчика по разрешению проблемных ситуаций. № 4.

Данчул А. Н., Новиков А. П. Представление знаний о предметной области на основе гносеологической модели. № 10.

Долинина О. Н., Кузьмин А. К. Применение методов технической диагностики для отладки баз знаний нейросетевых экспертных систем. № 2.

Евграфов П. М. Метод структурирования, представления и логического оценивания "неидеальных" знаний-решений. № 4.

Ермаков А. Е. Извлечение знаний из текста и их обработка: состояние и перспективы. № 7.

Ермаков А. Е., Плешко В. В. Семантическая интерпретация в системах компьютерного анализа текста. № 6.

Загорюлько Ю. А., Боровикова О. И. Информационная модель портала научных знаний. № 12.

Керимов С. Г. О модели онтологии предметной области, модели информационного поиска и коррекции запросов. № 4.

Колдаев А. И. Моделирование интеллектуальной системы поддержки принятия решений при управлении технологическим процессом. № 3.

Кязимов Т. Г., Махмудова Ш. Д. Система компьютерного распознавания людей по фотопортретам. № 1.

Михеева Т. И., Потапкин А. А., Демьяненко Р. В. Адаптивная кластеризация графа улично-дорожной сети. № 3.

Норенков И. П., Трудоношин В. А., Кузьмин А. А., Кузьмина И. А. Генетические методы с фрагментными кроссовером и макромутациями. № 1.

- Ронжин А. Л.** Сравнительный анализ и оценка моделей словаря для систем распознавания русской речи. № 1.
- Савченко В. В., Пономарев Д. А.** Оптимизация фонетической базы данных по группе дикторов на основе информационной теории восприятия речи. № 12.
- Сулейманов А. Ш.** Метод определения контекстных слов при анализе текста. № 7.
- Ходашинский И. А.** Идентификация параметров нечетких моделей типа сингтон на основе алгоритма роящихся частиц. № 6.
- Швецов А. Н., Сергушичева М. А.** Проектирование прикладных мультиагентных систем с использованием пакета DISIT. № 8.
- Щуревич Е. В.** Кластеризация знаний в системах искусственного интеллекта. № 2.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ (ОПТИМИЗАЦИЯ И ПРИНЯТИЕ РЕШЕНИЙ)

- Аввакумов В. Д.** Оптимальное размещение плоских объектов произвольной геометрической формы. № 5.
- Аникин В. И., Аникина О. В.** Табличное моделирование клеточных автоматов в Microsoft Excel. № 6.
- Антонов В. А.** Информационная технология оптимизации новым численным методом и построения на его основе полиномиальных степенных трендов с самоопределяющимися показателями степени. № 8.
- Воловиков В. В., Увайсов С. У.** Модель тепломассопереноса в конструкциях радиоэлектронных устройств. № 8.
- Воловикова Е. В., Увайсов С. Т.** Метод диагностирования радиоэлектронных устройств на основе комплексного электротеплового моделирования. № 10.
- Голубинский А. Н.** Методика проверки на адекватность математической модели речевого сигнала экспериментальным данным. № 12.
- Девятов Д. Х., Файнштейн С. И., Белявский А. Б., Торчинский В. Е.** Эвристическая оптимизационная модель для задач с ограничениями, возникающих при оперативном планировании листопрокатного производства. № 9.
- Димитриенко Ю. И., Захаров А. А.** Автоматизированная система для моделирования газовых потоков методом ленточных адаптивных сеток. № 6.
- Казаков П. В.** Кластерное расширение генетического алгоритма для решения многоэкстремальных задач оптимизации. № 8.
- Карпенко А. П., Овчинников В. А., Семенихин А. С.** Программная система PRADIS//FRONT для построения множества Парето в задаче многокритериальной оптимизации динамических систем с использованием параллельного генетического алгоритма. № 8.
- Карпенко А. П., Уторов А. Н., Федорук В. Г.** MPI-балансер загрузки многопроцессорной вычислительной системы для решения задачи многокритериальной оптимизации. № 6.
- Карпенко А. П., Федорук В. Г.** Один класс прямых адаптивных методов многокритериальной оптимизации. № 5.
- Колоколов А. А., Адельшин А. В., Ягофарова Д. И.** Решение задачи выполнимости с использованием метода перебора L -классов. № 2.
- Кухаренко Б. Г.** Анализ независимых компонент записей колебаний в технологии спектрального анализа на основе быстрого преобразования Фурье. № 9.
- Кухаренко Б. Г.** Предварительная обработка записей колебаний в технологии спектрального анализа на основе быстрого преобразования Фурье. № 2.

- Мезенцев Ю. А.** Оптимизация расписаний параллельно-последовательных систем в календарном планировании. № 6.
- Михайлов Б. М., Александров А. Е.** Решение граничных обратных задач теплопроводности на основе технологии порождающего программирования. № 11.
- Мухачева Э. А., Заминова А. Р.** Исследование эффективности эволюционных алгоритмов для решения задач дискретной оптимизации на примере задачи коммивояжера. № 10.
- Мухачева Э. А., Хасанова Э. И.** Гильотинное размещение контейнеров в полосе: комбинирование эвристических технологий. № 11.
- Нечаев В. В.** Концептуальное модельное представление задачи как системы. № 9.
- Опоков Ш. А.** Обобщение вычислений над полем комплексных чисел с исключением ошибок округления. № 6.
- Опоков Ш. А.** Применение модулярной арифметики с фиксированной точкой для ослабления влияния ошибок округления компьютерных вычислений. № 12.
- Поляков Б. Н.** Эффективный метод ранжирования независимых переменных и отбрасывания несущественных параметров при многофакторном статистическом анализе. № 11.
- Потапов А. П., Петров И. Б.** Моделирование высокоскоростных соударений методом сглаженных частиц (SPH). № 8.
- Рагманова И. Д., Коровкин С. Д., Железняк Н. В.** Информационная модель топливно-энергетического комплекса как основа анализа энергетической безопасности региона. № 9.
- Секаев В. Г.** Использование алгоритмов комбинирования эвристик при построении оптимальных расписаний. № 10.
- Филиппова А. С., Филиппов Д. В., Гильманова Н. А.** Задачи маршрутизации в транспортных логических системах: локальный поиск рациональных решений. № 2.
- Яковлев М. А., Чугунков И. В.** Повышение эффективности оценочных тестов для псевдослучайных последовательностей. № 2.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Волченков Е. Я.** Система комбинационной генерации тестов вход/выход: основные методы и алгоритмы. № 5.
- Вьюкова Н. И., Галатенко В. А., Самборский С. В.** Обобщение задачи программной конвейеризации для гнезд циклов. № 3.
- Губайдуллин И. И., Фрид А. И.** Метод экспериментальной оценки надежности программного обеспечения. № 5.
- Зуев А. С., Кучеров О. Б.** Модификация принципов работы с дочерними окнами программ, панелями инструментов, главными и контекстными меню. № 4.
- Зуев А. С., Петров Ю. И.** Описание модификации строки адреса проводника Windows Explorer. № 3.
- Князева М. А., Плохих С. А.** Концепция системы управления специализированного банка знаний о преобразованиях программ. № 5.
- Лаптев В. В., Морозов А. В.** Экспертная система оценки качественных характеристик проектной модели программного обеспечения. № 3.
- Макаренко В. И., Подольская Н. Н.** Полезные приемы интерактивного проектирования программного обеспечения модифицируемых систем управления. № 4.

- Пекунов В. В.** Процедуры с планированием повторного входа в языках высокого уровня при традиционном и параллельном программировании. № 8.
- Филиппов А. Н.** Метод нумерации значений и использование его результатов при оптимизации программ. № 4.

КОМПЬЮТЕРНАЯ ГРАФИКА

- Архипов О. П., Зыкова З. П.** Обеспечение идентичности результатов печати на различных цветных принтерах. № 3.
- Бимаков В. А.** Высокоскоростная обработка n -мерных пиксельных геометрических моделей. № 3.
- Кудряшов А. П.** Реконструкция трехмерных сцен городской обстановки. № 7.
- Левашкина А. О., Поршнев С. В.** Сравнительный анализ супервизорных критериев оценки качества сегментации изображений. № 5.
- Максименко-Шейко К. В., Толоч А. В., Шейко Т. И.** R-функции и аналитическое описание геометрических объектов, обладающих симметрией. № 7.
- Масалович А. А.** Численные методы детектирования и удаления геометрических искажений в изображениях текстовых документов. № 5.
- Сафин М. Я.** Геометрический процессор для определения видимости, теней и освещенности. № 11.

БАЗЫ ДАННЫХ И ЗНАНИЙ

- Белова Н. С.** Методика приоритетного автовосстановления встраиваемых баз данных. № 11.
- Горелов С. С.** Модели и алгоритмы для систем поиска в наборах документов. № 1.
- Куренков Н. И., Ананьев С. Н.** Критерий однородности матрицы и его использование в анализе многомерных данных. № 1.
- Левандовский В. И.** Обеспечение защиты программных приложений от нелегального доступа. № 11.
- Мокрозуб В. Г.** Таксономия в базе данных стандартных элементов технических объектов. № 11.
- Полищук Ю. В., Черных Т. А.** Моделирование подсистем хранения информации, ориентированных на хранение квантизированных объектов. № 1.
- Редреев П. Г.** Построение табличных приложений со списочными компонентами. № 5.
- Соколов А. В.** Методика оценки максимально возможных значений показателей эффективности поиска текстовой информации. № 5.
- Сурпин В. П.** Разработка подсистемы ведения классификаторов корпоративной информационной системы. № 11.
- Туманов В. Е.** Предметно-ориентированные системы научной осведомленности. № 5.
- Шарашов В. В.** Методика выбора способа отображения наследования объектов в реляционную базу данных. № 5.

WEB-ТЕХНОЛОГИИ

- Асратян Р. Э.** Служба синхронизации процессов в сети. № 12.
- Владыкин А. А., Шальто А. А.** Непроцедурный текстовый язык описания автоматных обработчиков XML-документов и его применение. № 12.
- Жусов Д. Л., Комашинский В. В.** Варианты реализации модуля фильтрации потока запросов к Web-серверу. № 1.
- Пшинник К. В., Козицын А. С., Елкин Е. В.** Выделение значимой информации из Web-ресурсов. № 8.
- Тарнавский Г. А., Алиев А. В., Анищик В. С., Тарнавский А. Г., Жибинов С. Б., Чесноков С. С.** Информационные тех-

нологии и проблемы создания Центра компьютерного моделирования в Интернете. № 8.

- Чеснавский А. А.** Практическое применение алгоритма семантического анализа изменений в HTML-документах. № 1.

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ СИСТЕМЫ И ОБРАБОТКА СИГНАЛОВ (КОДИРОВАНИЕ И ОБРАБОТКА СИГНАЛОВ)

- Алексеев В. Е., Соловьев А. Н.** Многоантенные GPS-системы с дециметровой точностью позиционирования. № 4.
- Асадов Х. Г., Чобанзаде И. Г.** Информационный метод радиометрической коррекции и калибровки бортовых спектрометров в ходе полета. № 9.
- Гребнов С. В.** Разработка и реализация двухуровневого метода голосового управления на основе скрытых марковских моделей. № 9.
- Дворников С. В., Жечев А. Г.** Демодуляция сигналов на основе обработки их модифицированных частотно-временных распределений. № 4.
- Мальхина Г. Ф., Меркушева А. В.** Восстановление формы сигналов по измерительной информации об их смеси. Концепции, модификации алгоритмов и приложения. № 9.
- Медведева Е. В.** Адаптивная нелинейная фильтрация цветных видеоизображений. № 11.
- Сулейманов А. Ш.** Алгоритмы сжатия данных, основанные на применении логических шкал позиционного кодирования. № 11.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

- Кораблин М. А., Салмин А. А., Бедняк О. И., Таев С. С.** Категориальный анализ и оценка поведения клиентов для прогнозирования рыночных отношений. № 11.
- Куперин Ю. А., Счастливцев Р. Р.** Прогнозирование финансовых временных рядов с помощью комитетов искусственных нейронных сетей. № 3.
- Лиманова Н. И., Седов М. Н.** Метод автоматизированного поиска персональных данных на основе нечеткого сравнения. № 11.
- Рзаев Р. Р., Алиев Э. Р.** Оценка профессиональных качеств служащих компании методом нечеткого логического вывода. № 11.
- Серков Л. А.** Синергетические аспекты моделирования инновационных процессов. № 3.
- Соловецкий А. С.** Мобильные информационные технологии в маркетинге и рекламе. № 3.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СОЦИАЛЬНЫХ И ЭКОЛОГИЧЕСКИХ СИСТЕМАХ

- Бождай А. С.** Комплексная инфраструктура территории: методы и модели информационного мониторинга. № 9.
- Дурнев Р. А.** Система информирования и оповещения населения: функции и структура. № 1.
- Переварюха А. Ю.** Нелинейные модели и особенности оптимизации в задаче системного анализа динамики популяций. № 1.
- Филиппова А. С., Брусиловский Д. П.** Прогнозирование в ситуации непостоянного спроса для оптимального управления складскими запасами производственно-сборочных компаний. № 9.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Зюбин В. Е.** Использование виртуальных объектов для обучения программированию информационно-управляющих систем. № 6.
- Карданова Е. Ю., Карпинский В. Б.** Использование эксперимента на модели Раша для выявления недостоверных результатов педагогического тестирования. № 6.
- Норенков И. П., Соколов Н. К.** Синтез индивидуальных маршрутов обучения в онтологических обучающих системах. № 3.
- Пономарев Д. А.** Двухэтапный алгоритм обучения речи слабослышащих на основе информационной теории восприятия речи. № 9.
- Рыбаков А. Е.** Моделирование структуры учебных курсов и подсистемы навигации для систем управления обучением, поддерживающих стандарт SCORM. № 3.
- Сарафанов А. В., Комаров В. А.** Разработка математической модели многопользовательского режима функционирования аппаратно-программных комплексов с удаленным доступом. № 3.
- Штрик А. А.** Индикаторы информационной грамотности. № 9.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ И БИОЛОГИИ

- Анищенко В. С., Булдакова Т. И., Довгалевский П. Я., Лифшиц В. Б., Гриднев В. И., Суятин С. И.** Концептуальная модель виртуального центра охраны здоровья населения. № 12.
- Воробейчикова О. В.** Технология построения структуры усвоения учебного материала по результатам тестирования. № 2.
- Дьяконов В. П., Хотова Ф. А.** Матричная система МАТ-LAB в биоинформатике. № 6.
- Исаева О. С.** Унифицированная информационная модель описания медицинских услуг. № 2.
- Кожевников М. А., Марчук Ю. В., Хамидулина О. Н., Монтиле А. И., Погосян И. А.** Разработка средства поддержки диагностики ортопедической патологии на основе дискриминантного анализа клинико-anamnestических данных. № 6.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Агаев Н. Б.** Диагностирование предаварийного состояния систем газоснабжения на основе теории нечетких множеств. № 10.
- Воронова Л. М.** Моделирование оптимальных планов земледелия с применением адаптированных пакетов технологий. № 5.
- Гермашев И. В., Васильева В. Д., Дербишер В. Е., Дербишер Е. В., Силина А. Ю.** Применение теории нечетких множеств для решения задач химической технологии. № 5.
- Дорожко В. М.** Информационная технология идентификации начального этапа разгона морского судна. № 10.
- Коваленко С. Н.** Методика определения допустимых сбросов дренажных вод с учетом стохастического процесса на основе математического моделирования концентрации биогенных загрязняющих веществ в малых реках (на примере Двинско-Печерского бассейна). № 11.

- Подольская Н. Н.** Проектирование человеко-ориентированного программного обеспечения отображения воздушной обстановки. № 11.
- Сердобинцев С. П., Коломейко Ф. В.** Применение информационных технологий в прогнозировании распределения объектов рыбопромыслового лова. № 3.

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

- Каргапольцев С. К., Лашук Н. В.** Система поддержки принятия решений для обеспечения автоматизации управления вузом. № 6.

ДИСКУССИОННЫЙ КЛУБ

- Вяткин В. Б.** Синергетический подход к определению количества информации. № 12.
- Котляров И. Д.** Сетевая публикация результатов диссертационных исследований. № 7.
- Левин В. И.** Что такое интервальная задача математического программирования. № 2.

ОБМЕН ОПЫТОМ

- Федоренко О. В.** Статистический подход к определению приоритета критериев для рейтингового оценивания научных журналов методом анализа иерархий. № 4.

Указатель приложений к журналу "Информационные технологии", опубликованных в 2009 году

- Валихметова Ю. И., Мухачева Э. А., Филиппова А. С., Гильманова Н. А., Карипов У. А.** Мультиметодная технология ортогональной упаковки и ее применение в задачах транспортной логистики. № 12.
- Васенин В. А., Афонин С. А., Козицын А. С.** Автоматизированная система тематического анализа информации. № 4.
- Васенин В. А., Инюхин А. В., Шевелев М. В.** Вычислительный Grid-полигон: состояние, идеи, решения. № 7.
- Кудинов Ю. И.** Системный подход к нечеткому моделированию сложных производственных систем. № 5.
- Кузин Е. С.** Конструктивная семантика. № 11.
- Ледокол.** Страницы биографии Анатолия Ивановича Китова. № 3.
- Мухачева Э. А., Валеева А. Ф., Сиразетдинова Т. Ю., Сиразетдинов Т. М.** Автоматизация проектирования гильотинного раскроя с обходом дефектных областей на базе эволюционных алгоритмов. № 2.
- Норенков И. П.** Содержание и информационная поддержка подготовки специалистов в области информатики и вычислительной техники. № 9.
- Трахтенгерц Э. А.** Компьютерные технологии информационного управления в конфликтных ситуациях. № 1.
- Шилов В. В.** Игры, в которые играли автоматы. Исторические очерки. № 8.
- Штрик А. А.** Использование информационно-коммуникационных технологий для экономического развития и государственного управления в странах современного мира. № 6.
- Штрик А. А.** Электронные технологии в деятельности органов государственной власти России: анализ и перспективы развития. № 10.

CONTENTS

Zagorulko Yu. A., Borovikova O. I. *The Information Model of Scientific Knowledge Portal* 2

The paper presents an information model of scientific knowledge portal which provides content-based access to the scientific information resources. The information model integrates models of subject and problem domains of the portal, and describes types of represented information. Based on the ontology, internal storages of the portal data are constructed and management of its information content, as well as navigation and search, are organized.

Keywords: ontology, knowledge portal, information model, scientific information resources.

Savchenko V. V., Ponomaryov D. A. *Phonetic Database Optimization Based on Information Theory of Speech Perception* 7

The problem of forming and optimization of phonetic database (PDB) by selected speakers group is defined and solved. Based on the information theory of speech perception criterion of the minimal information mismatch of distributions between united and personal phonetic databases is offered and substantiated. The experimental investigation results are given.

Keywords: speech, phonetic database, information theory of speech perception, information approach.

Vashkevitch N. P., Bektashev R. A. *The Formalization of the Synchronization Processes Algorithm when Dispatching Tasks in Multiprocessor Systems with the Use of Rendezvous Mechanism* 12

The article deals with the formal presentation of algorithms for the synchronization of interacting processes acting in parallel computer systems when dispatching tasks in the queue to the FIFO processor type. In these synchronization algorithms method "rendezvous" and primitive "monitor" are applied. To formalize, the logic of the nondeterministic automata is used. The equations that describe the procedures for client and server parts of the synchronization algorithm, bearing in mind the discipline to ensure priority and mutual exclusion of processes when choosing the executable tasks (processors).

Keywords: multiprocessing systems, scheduling of tasks, synchronisation of processes, not determined state machine, algorithm "hairdressing salon work", of rendezvous mechanism.

Serikov D. A. *Mathematical Modeling of the Process of Dispatching Jobs in a Distributed Computing Environment* 17

The paper discusses approaches to mathematical modeling of the division of resources between jobs of the various users which arrive to services in a distributed computing environment, based on the methodology GRID. Analyzed a model of the system of dispatching, which are used in the instrumental complex GridWay.

Keywords: grid, dispatching, modeling.

Kovalenko O. N. *Influence of Method of Treatment of Turns on Quality of Maintenance of Traffic of Multiservis Communication Networks* 25

At construction of multiservice communication networks of one of the main tasks granting of demanded quality of service to various kinds of the traffic is. To solve the given problem it is possible by means of classification of arriving streams and flexible distribution of resources between turns. In work mathematical models of the most widespread ways of division of throughput of the channel between competing streams — PQ and CBWFQ for system of mass service $M/M/N:n$ are offered.

Keywords: service, channel, quality, turn, priority.

Volosatova T. M., Belomoytsev D. E. <i>Mobile Devices Piconet Building Technologies and Methods Libraries</i>	28
Methods of secure message passing channels design are described. Mobile device wireless connection channel model is introduced. Client-server cooperation software realization is developed.	
Keywords: design, automation, wireless, mobile.	
Andreev O. O. <i>Integration of Access Control Models Defined in Special Language</i>	29
Article presents access control models description language. Issues of automated analysis of models, described with this language are studied.	
Keywords: information security, access control, integration of access control models.	
Ushmaev O. S. <i>Biometric Applications for Integrated Information Security</i>	34
We consider applications of biometrics to integrated information security. An outline of biometric technologies and their advantages is presented. We analyze typical solutions for different IT security applications: physical access, identifications in OS and networks, data protection. We proposed new approach to integration of short listed applications basing on the single biometric identification.	
Keywords: biometric identification, user authentication, access control.	
Vladykin A. A., Shalyto A. A. <i>Proposal and Applications of a Declarative Language for Automata-Based XML Handlers</i>	38
A declarative language for XML document handlers is proposed. The handlers are intended to be used together with the standard SAX parser. They are implemented using automata, which simplifies their structure and understanding.	
Keywords: XML, SAX-handlers, automata.	
Asratian R. E. <i>A Service for Parallel Processes Synchronization in Network Environment</i>	45
The organizational principles of a new Internet service oriented to parallel processes synchronization in global network environment are described. The service can be viewed as a sort of "network generalization" of conventional interprocess interaction and synchronization tools, which all modern operating systems provide for (mutexes, critical sections, conditional variables etc.) and can be used in distributed information systems development.	
Keywords: distributed systems, Internet services, process synchronization.	
Otsokov Sh. A. <i>Application of Fixed Point Modular Arithmetic for Reducing Influence of Round-off Errors of computer computation</i>	50
A fixed point modular arithmetic for an exception of catastrophic loss of accuracy of computer computation with very different from each other numbers is offered. The effect of catastrophic loss of accuracy is shown on an example of definition of scalar product of two vectors and discrete transformation of Fure.	
Keywords: modular arithmetic, modular format, working positive range, working negative range.	
Golubinskiy A. N. <i>Method of Mathematical Model Verification of a Speech Signal to Experimental Data</i>	54
The majority of used measure of concordance of mathematical model of a speech signal to experimental data is proposed and analyzed. The method of mathematical model verification of a speech signal with using exact statistical criterion is designed.	
Keywords: mathematical model, adequacy, speech signal, measure of concordance, error of model, statistical criterion.	
Anitshenko V. S., Bouldakova T. I., Dovgalevsky P. Ya., Lifshits V. S., Gridnev V. I., Suyatinov S. I. <i>Conceptual Model of Virtual Centre of Public Health Services</i>	59
There are considered the substantive provisions, principles, conditions and mechanisms of realization in the form of the virtual centre of the new concept in public health services — the personalized	

medicine. The virtual centre unites all making elements of system of public health services on the basis of the general information field, providing information gathering, the deep analysis and an exchange of great volumes of the data. In this concept the important problem is creation of virtual human physiology. The virtual "copy" of the patient constructed by means of mathematical models of elements and subsystems of an organism, describes activity of physiological subsystems of the person and represents its virtual physiological image.

Component of the general virtual model of human physiology is the computer model of biosystem "heart — vessels — lungs". The method of construction of this model on the basis of structurally-parametrical identification and its use for an estimation of a functional state is offered.

Transition to the virtual form of the organization of public health services assumes stage-by-stage realization of separate projects. A priority problem of the first stage is creation of methods and means of express monitoring and an estimation of a state of health of the person. The prototype of the virtual centre of health protection is offered.

Keywords: personalised medicine, monitoring, estimation of functional state, virtual physiological image, biosignals, system "heart — vessels — lungs", model equations.

Makhortov S. D. *An Integrated Environment for the Logical Programming LPExpert*. 65

In a number of the earlier publications the author proposed an algebraic apparatus (the theory of *LP*-structures) allowing one to formalize the production logical inference on the basis of the theory of lattices and relations. This apparatus can be applied to the automated study of the properties of knowledge bases on the basis of formal models, including their verification and optimization. In the present paper the functionality of the software package *LPExpert* created by the author, which is intended for the development of the production systems and which contains realization of *LP*-structures, is briefly presented.

Keywords: production system, algebraic model, knowledge management, computer implementation.

Kiryanov A. A., Sirotkin V. Yu., Sirotkin Yu. V., Smirnova A. A. *Software for the Computer Aided System for Environmental Monitoring at a Plant* 67

Results of development of the program complex for working in the automatical system of ecological monitoring at an enterprise are described in the article. The complex allows collecting information from detectors of the environmental condition to the integrated database and informs the operator about excess of admissible concentration of harmful substances and about tendencies of changing of ecological situation.

Keywords: the program complex, ecological monitoring at an enterprise.

Vyatkin V. B. *Synergetic Approach to Definition of Information Quantity* 68

New approach is stated to definition of information quantity, where for information datum are taken about finite set how uniform whole, and information measure is average length of integrative code of elements.

Keywords: negentropy, information quantity, reflection, finite set, element, sign, integrative code.

Адрес редакции:

107076, Москва, Стромынский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *О. А. Ефремова*.
Корректор *Е. В. Комиссарова*

Сдано в набор 07.10.2009. Подписано в печать 18.11.2009. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 9,8. Уч.-изд. л. 11,07. Заказ 994. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"
142110, Московская обл., г. Подольск, ул. Кирова, 15