ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

5(165)
2010

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

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.

УДК 004.8;004.3

Л. Г. Комарцова, д-р техн. наук, проф., зав. каф.,
e-mail: lkomartsova@yandex.ru,
Д. С. Кадников, аспирант,
И. В. Ковалев, аспирант,
Калужский филиал МГТУ им. Н. Э. Баумана

Особенности построения гибридных интеллектуальных систем обработки информации

Исследованы принципы создания интеллектуальных гибридных систем, обеспечивающих решение разнообразных прикладных задач в условиях неполноты и нечеткости исходной информации. Показано, что дальнейшее направление исследований в этой области связано с созданием эволюционных, постоянно развивающихся динамических интеллектуальных систем, работающих в режиме on-line и подстраивающихся под конкретную решаемую задачу.

Ключевые слова: *эволюционные алгоритмы, эволюционное программирование, эволюционные системы, нейронные сети, гибридные системы, экспертные системы*

Введение

В последние годы наблюдается рост числа успешных примеров использования гибридных интеллектуальных систем (ИС) в различных прикладных областях, таких как медицинская диагностика, распознавание речи и естественных языков, создание мобильных роботов, мониторинг и контроль производственных процессов, финансовые приложения. Гибридные системы (ГС, HIS), работающие на основе принципов объединения нескольких методов представления и обработки информации, позволяют получать значительно более лучшие результаты решения по сравнению с ИС, использующими единственный метод для тех же проблем [2—4, 7]. Однако проблемы создания и использования гибридов для конкретных приложений все еще не решены. Остается много вопросов, связанных с тем, на каком уровне проводить объединение разных интеллектуальных технологий, какие гибриды являются наиболее перспективными, как учесть динамику изменения среды функционирования ИС и т. д. [3, 13, 18]. Для решения этих и других проблем рассмотрим особенности наиболее известных интеллектуальных технологий и возможности их интеграции в гибридные ИС (Hybrid Intelligence System — HIS).

Этапы развития наиболее часто применяемых на практике интеллектуальных технологий (ИТ) во времени представлены на рис. 1. Анализ периодов развития ИТ показывает, что, с одной стороны, появляются новые технологии, призванные решать все усложняющиеся задачи теории и практики, а с другой — происходит объединение ИТ для более эффективного решения традиционных проблем.

Область исследований для развития моделей, методов и основных технологий для представления и обработки знаний и построения интеллектуальных систем, основанных на знаниях, называется *инженерией знаний* [2].

Основные задачи инженерии знаний сводятся к следующему:

- *представление знаний* — процесс приведения знаний о существующей проблеме к некоторой известной схеме с помощью методов инженерии знаний;
- *вывод* — процесс приведения в соответствие текущих фактов из некоторой проблемной области существующим знаниям об этой области или выводу новых фактов;
- *обучение* — процесс получения новых знаний в результате обобщения текущей информации: а) обучение на примерах; б) путем выполнения ряда правил; в) на основе инструкций извне;
- *обобщение* — процесс приведения в соответствие неизвестных входных данных имеющимся знаниям о проблеме для получения лучшего решения. Это переход от частного описания объекта к общему;
- *взаимодействие* между пользователем и ИС, важное для адаптации ИС к новым ситуациям.

Рис. 1. Развитие интеллектуальных технологий во времени

Взаимодействие между подсистемами ИС является главной характеристикой для распределенных ГС, в которых каждый модуль принимает участие в решении проблемы (агентно-ориентированный подход);

- *объяснение* — это желаемое свойство ИС, связанное с трассировкой процедуры получения решений для понимания того, как оно было получено;
- *тестирование* — проверка работоспособности системы. Полученные с помощью ИС результаты сравниваются с результатами, полученными экспертами или другими ИС;
- *адаптация* — процесс изменения системы во время ее функционирования в динамически изменяющейся среде.

Рассмотрим, как эти задачи решаются в каждой из интеллектуальных технологий.

Экспертные системы

Экспертные системы (ЭС) являются знание-ориентированными системами и при функционировании используют знания экспертов в определенной прикладной области. ЭС содержат средства для представления и накопления знаний, получения новых знаний из существующих баз данных на основе логического вывода, принимают решения и выдают рекомендации по поставленной проблеме, взаимодействуют с пользователем (часто на ограниченном естественном языке), объясняют свое "поведение" и принятое решение [10].

ЭС используются почти во всех областях деятельности человека: на производстве, в науке, образовании, медицине, сельском хозяйстве, бизнесе, финансах и т. д. Основываясь на технологии баз знаний, ЭС могут быть быстро и дешево реализованы для решения конкретной прикладной задачи, могут обновлять базу знаний при появлении новых фактов и ситуаций. При этом необходимо различать две категории людей, имеющих отношение к ЭС: эксперты, которые аккумулируют свои знания в базу знаний, и пользователи, которые эти знания используют [10, 12].

Несмотря на то, что существует довольно много методов создания ЭС, которые давно и успешно используются, все еще остаются главные проблемы, заключающиеся в следующем [7, 10]:

- Как извлекать знания из экспертов (дефицит которых с каждым годом ощущается все острее)?
- Как извлечь знания из огромной массы предварительно собранных данных о решаемой проблеме?
- Как представлять неполные, неопределенные, искаженные и противоречивые данные и знания?
- Как моделировать человеческое мышление?

Решение этих проблем возможно при использовании других интеллектуальных технологий, таких как нечеткие логические системы, работающие на основе нечеткой логики (НЛ), и нейронные сети (НС) [13, 14].

Нечеткие системы

Одним из путей представления неточных данных и знаний, которыми обычно оперирует человек при решении трудных задач, является использование нечетких правил вместо четких правил ЭС [11]. В НЛ для получения решения используются нечеткие базы правил и нечеткий вывод. Нечеткие правила могут оперировать неопределенными, искаженными и противоречивыми данными и знаниями, полученными из различных источников или на основе анализа экспериментальных данных, из опыта многих исследователей, работающих в конкретной прикладной области.

Теоретической базой НЛ является теория нечетких множеств, предложенная Lotfi Za-deh. Основная цель использования НЛ — это моделирование аспектов человеческого мышления. НЛ могут быть менее точными, чем ЭС, но более приближены к человеческому опыту принятия решений. Хотя человек обычно оперирует нечеткими терминами, они в большей степени соответствуют реальной ситуации и позволяют нам глубже понять ее, нежели на основе описания количественными соотношениями.

НЛ определяются тремя главными компонентами [10—12, 14]:

- нечеткими входными и выходными переменными, определяемыми нечеткими величинами;
- множеством нечетких правил;
- механизмом нечеткого вывода.

Нечеткие правила оперируют нечеткими величинами, такими как "высокий", "очень низкий", "средний" и т. д. Эти нечеткие концепты обычно представляются с помощью функции принадлежности, которая определяет степень, с которой некоторая величина из проблемной области (обычно называемой универсумом) принадлежит некоторому нечеткому концепту. Процедура преобразования четкой величины в нечеткую называется *фаззификацией* [11, 12, 14]. Методы нечеткого вывода, базирующиеся на нечеткой логике, позволяют на основе использования четких или нечетких входных величин получать четкие или нечеткие выходные величины, которые с помощью процедуры дефаззификации преобразуются в четкие.

Число областей применения нечеткой логики постоянно расширяется: бытовая техника (стиральные машины, холодильники, пылесосы), производство (например, управление доменной печью, самолетом), робототехника, системы поддержки принятия решений и т. д. [8, 10]. Секрет успеха НЛ в различных областях заключается

в простоте их программирования, легкости обслуживания, робастности, дешевизне.

Нейронные сети

В процессе своего развития ЭС двигались в направлении разработки новых методов представления и обработки знаний, которые бы приближались к тем, которыми пользуется человек при решении сложных задач. Одной из таких моделей является искусственная нейронная сеть (НС), в которой реализованы только некоторые особенности биологических систем [8, 9, 16].

НС состоит из обрабатывающих элементов (искусственных нейронов), соединенных между собой определенным образом (способ соединения задает топологию НС и определяет ее свойства). Искусственные нейроны получают входные сигналы, которые имитируют электрические импульсы, получаемые дендритами биологических нейронов от других нейронов. Выход искусственного нейрона соответствует сигналу, посылаемому биологическим нейроном со своего аксона на входы других нейронов, осуществляя их активацию (возбуждение или торможение).

НС как вычислительная модель характеризуется четырьмя главными компонентами:

- типом нейрона (способ агрегации входов, вид функции активации);
- топологией, называемой также коннекционистской архитектурой (число слоев в НС, число нейронов в каждом слое, организация связей между нейронами);
- алгоритмом обучения;
- алгоритмом распознавания.

Наиболее важным свойством НС является ее *способность к обучению* [9, 13] на известных примерах (образцах), при этом используются, как и у человека, два типа памяти: текущая (кратковременная) память (активация определенных нейронов — возбуждение или торможение — в ответ на входное воздействие) и долговременная (матрица весовых связей между нейронами обученной НС), своего рода знания, полученные в процессе обучения.

Одной из главных особенностей нейросетевой модели, позволяющей рассматривать НС как перспективную интеллектуальную технологию, является ее возможность распознавания новых образов (или примеров) из проблемной области, отличных от тех, на которых обучалась НС. Эта особенность — *обобщение*, т. е. при поступлении нового входного вектора обученная НС будет формировать наиболее правдоподобный выход в соответствии с содержимым долговременной памяти. Другими, не менее важными достоинствами НС являются *естественный параллелизм* в обработке информации, основанный на одновременном возбуждении — торможении многих нейронов НС, *робастность*, заключающаяся в нивелировании работы некоторых "ложных нейронов" на фоне правильной работы всей сети, *адаптация* к новым данным, *фильтрация* шума в данных, возможность работы с неполными, неточными, искаженными данными и знаниями [8, 9, 13].

Таким образом, НС могут быть использованы в качестве моделей для имитации отдельных свойств человеческого мышления. В настоящее время они применяются в системах распознавания речи, образов, текста, для поиска релевантной информации в сети Internet и базах данных, в системах обнаружения атак на компьютерные сети, в обучающих системах, в робототехнических комплексах и т. д. Разработка новых типов моделей НС позволит приблизиться к пониманию проблем функционирования мозга человека и расширить область применения этих моделей.

Системы, основанные на прецедентах

Рассматриваемая интеллектуальная технология (Case Based Reasoning — CBR) получила развитие в середине 80-х годов прошлого столетия и основывается на использовании исторической информации, т. е. ранее полученных результатов решения некоторой проблемы для лучшего решения новых проблем, аналогичных предыдущей. В этой технологии реализуются аспекты человеческого мышления в том смысле, что мы при решении сложной текущей задачи, как правило, ссылаемся на прошлый опыт. В CBR сохраняются результаты ранее решенных проблем, и при появлении новой проблемы выявляются ее отличия от ранее решавшихся. Анализируется описание проблемы для сопоставления с ранее имевшимся прецедентом (случаем). Если найденный в базе прецедентов случай сопоставим с текущей проблемой, то сразу выдается решение; если он не является достаточно близким, пытаются его изменить и предложить решение вручную. Если снова решение не адекватно, эксперт определяет решаемую проблему как новый случай с запоминанием его в базе случаев вместе с предложенным решением для будущего использования.

CBR могут быть использованы самостоятельно или в составе другой интеллектуальной технологии обычно в тех случаях, когда эксперту трудно формулировать правила или когда база правил слишком велика. Область частого применения CBR — юриспруденция, медицина, другие диагностические системы, богатые историями прецедентов.

Генетические алгоритмы

Генетические алгоритмы (ГА) относятся к интеллектуальной технологии, в основе которой ле-

жит моделирование элементов природной эволюции. ГА были введены Дж. Холландом в 1975 г. и развиты другими последователями. Природное разнообразие биологических видов огромно. Как же в этом случае природа решает проблемы совершенствования видов? Первый ответ на этот вопрос дал Ч. Дарвин в теории эволюции: выживает наиболее приспособленный к среде обитания вид. Вот почему в основу функционирования ГА положена концепция теории наследственности Дарвина, связанная с естественным отбором. Большинство важнейших терминов, используемых в ГА, являются аналогами терминов, используемых для объяснения эволюционных процессов:

- ген — основная единица, определяющая свойства некоторой индивидуальности;
- хромосома — строка генов, используемая для представления индивидуальности, другими словами, одно из возможных решений проблемы;
- популяция — множество хромосом;
- кроссинговер — операция для скрещивания разных индивидуальностей в целях получения новых индивидуальностей (потомков);
- мутация — случайное изменение гена в хромосоме;
- фитнесс (функция качества Fit) — критерий оценки каждой индивидуальности;
- отбор — подбор хромосом для выполнения операции скрещивания;
- селекция — процедура для выбора индивидуальностей текущей популяции в следующую популяцию на основе функции фитнесса; на каждой стадии развития популяции лучшее решение сохраняется.

Для ГА наиболее важными являются следующие четыре параметра [15, 22]:

- схема кодирования решения: структура хромосомы, используемые способы кодирования хромосомы (двоичный код, вещественные числа и т. д.);
- размер популяции, определяющий, сколько возможных решений необходимо хранить для оптимального развития популяции;
- операция кроссинговера — как комбинировать "старые" индивидуальности и продуцировать новые, более перспективные;
- операция мутации — когда и каким образом необходимо "локально", на уровне генов изменять хромосому.

Основные особенности ГА сводятся к следующему [17, 19, 20]:

- ГА — это эвристические методы для решения сложных задач. В противоположность исчерпывающему исследованию ГА не рассматривают все варианты, чтобы выбрать лучшее решение. Поэтому может быть найдено не идеальное решение, а близкое к нему за ограниченное время;
- ГА обладают адаптационными возможностями, т. е. способны обучаться и накапливать знания и факты "с нуля". Для успешной работы алгоритма трудной проблемой является выбор функции качества (Fit), которая будет отбирать в следующую популяцию только лучшие решения;
- ГА обладают высоким параллелизмом за счет одновременного развития многих популяций. Поэтому каждая ветвь дерева решений может быть исследована параллельно с другой, что значительно сокращает время решения проблемы;
- ГА могут быть использованы в качестве обучающих модулей в ЭС или выступать в качестве самостоятельных интеллектуальных систем обработки информации.

Одним из расширений класса ГА является генетическое программирование (ГП), предложенное J. Koza в 1992 г. Под ГП понимается применение генетической модели в пространстве программ, при этом все операции, связанные с изменением вида, осуществляются не над строками, а над деревьями, представляющими фрагменты компьютерных программ. Каждая программа в популяции оценивается с помощью функции качества, например, с помощью ошибки, с точностью до которой происходит решение задачи. В качестве генетических операторов используются такие же, как и в ГА: отбор, кроссинговер, мутация, селекция.

Основная цель ГП — сформировать компьютерную программу, которая наилучшим образом будет решать поставленную проблему. Начальная популяция представляет собой совокупность случайным образом сформированных программ, составленных из определенного числа функций и терминальных символов с учетом рассматриваемой предметной области. Эти функции могут включать в себя математические операции, стандартные операторы программирования, стандартные математические функции, логические и специальные функции. Применение генетических операторов к популяции позволяет на каждом шаге (в каждом поколении) отбирать лучшие фрагменты программ с учетом заданной функции Fit, что в конечном итоге приведет к созданию лучшей из возможных программ.

В настоящее время ГП успешно применяется для решения задач символьной регрессии, анализа данных, оптимизации, оптимального управления сложными механизмами и т. д. [13, 17].

Эволюционные стратегии отличаются от ГА и ГП в нескольких направлениях: они оперируют не хромосомами (в бинарной кодировке), а вещественными величинами; популяции могут описы-

ваться статистическими параметрами; эволюция популяции осуществляется на основе применения операции мутации. Использование эволюционных стратегий позволяет создавать эволюционные ИС, которые могут пополнять свои знания в процессе функционирования в определенной среде, адекватно реагировать на изменения этой среды в реальном масштабе времени [15, 17, 21, 24].

Сравнительные возможности интеллектуальных технологий

В таблице представлены результаты сравнения технологий, чаще всего используемых на практике при решении ими задач инженерии знаний.

Анализируя таблицу, можно сделать вывод, что НС является наилучшей технологией для решения задач обучения, адаптации и обобщения, ЭС и нечеткие системы обладают хорошими объяснительными возможностями, что создает предпосылки для объединения технологий в целях компенсации недостатков и усиления достоинств каждой из технологии при решении сложных задач [2, 3, 6, 7, 14].

Под гибридной интеллектуальной системой (ГС) будем понимать систему, в которой для решения задачи используется более одного метода имитации интеллектуальной деятельности человека [3, 7, 13, 14]. Интеграция методов, с одной стороны, дает возможность использовать индивидуальную силу каждого из методов для решения специфических частей задачи, что позволит создать более эффективные модели представления и обработки знаний. С другой стороны, гибридный подход основывается на том, что только синергетическая комбинация интеллектуальных технологий может достичь полного спектра когнитивных и вычислительных возможностей, реализуемых в компьютерных моделях интеллектуальных систем.

Существуют, по крайней мере, две главные причины, по которым необходимо использовать именно гибридные системы [1]:

- некоторые требования для решения проблем искусственного интеллекта (ИИ) не могут быть принципиально выполнены на основе единственного подхода;
- для решения сложных проблем ИИ создаваемые модели также не могут быть реализованы с помощью одного какого-то метода.

Сравнительные возможности интеллектуальных технологий

Задачи	ЭС	НЛ	НС
Представление знаний	Структуризация в виде правил	Структуризация в виде правил	Неструктурное обученная НС
Вывод	Точный	Приближенный	Приближенный
Обучение	Среднее	Нет	Очень хорошее
Обобщение	Слабое	Очень хорошее	Очень хорошее
Взаимодействие	Хорошее	Хорошее	Хорошее
Объяснение	Очень хорошее	Очень хорошее	Слабое
Тестирование	Очень хорошее	Очень хорошее	Среднее
Адаптация	Слабая	Слабая	Хорошая

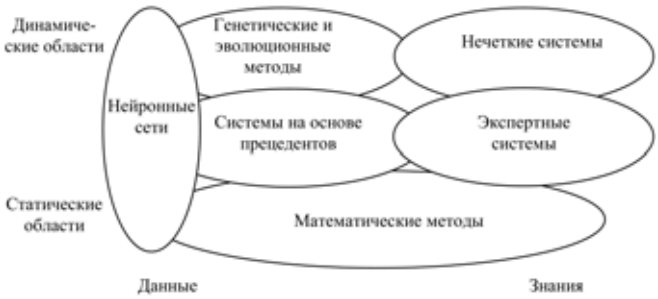

Рис. 2. Области использования интеллектуальных технологий

По мере социального и технического развития общества появляются новые требования, которые приходится учитывать при решении сложных интеллектуальных задач, например:

- представление точных и неточных, определенных и неопределенных знаний о проблеме в одной системе;
- извлечение знаний из данных;
- учет прошлого опыта для принятия решений при возникновении новых ситуаций;
- необходимость адаптации в поведении систем при изменении характеристик среды;
- терпимость к неточности;
- робастность;
- возможность расширения выполняемых функций;
- необходимость объяснения результатов решений;
- возможность получения альтернативных решений.

Естественно, что одного метода для удовлетворения всем этим требованиям не существует. Выбор методов для обработки исходной информации зависит от особенностей решаемых задач, от числа количественных и качественных параметров, описывающих проблему, от уровня проработанности задачи. Поэтому необходимо определять условия применимости каждого из методов, а также алгоритмы, позволяющие адаптировать их к решению конкретных задач проблемной области.

Обзор информационных технологий, использующихся в современных интеллектуальных системах (рис. 2), показывает, что в статических и динамических проблемных средах эффективными являются определенные методы [10—12, 16, 19, 20]. Например, в некоторых случаях наилучшее решение может быть получено путем использования эволюционных методов (в частности, на основе генетических алгоритмов) на всех этапах поиска рационального решения, что позволяет перейти

от моделей представления и использования знаний с жесткими связями к моделям с динамически меняющейся структурой в зависимости от решаемой задачи [15, 16, 19, 20, 24].

Классификация гибридных систем

Существует довольно большое число публикаций по проблемам построения ГС.

Цели интеграции отдельных технологий в HIS и основная терминология впервые были введены Bezdek J. в [5]. Предложенная им модель интеграции отдельных технологий базируется на различных уровнях интеллектуальной активности компонентов, объединяемых в иерархическую систему. Главное отличие введенной модели (рис. 3) от других ГС состоит в увеличении сложности решаемых задач при переходе от низшего уровня, соответствующего вычислительному интеллекту, к более высокому уровню искусственного интеллекта, а затем к биологическому, моделирующему человеческий интеллект. Этот рисунок показывает взаимосвязь уровней при обработке информации. В зависимости от имеющейся информации и решаемых задач активизируется определенный уровень. Вычислительный интеллект определяет уровень обработки числовых данных, ИИ основан на процедурах обработки символов и данных с использованием правил и нечисловых данных. Биологический интеллект обрабатывает сенсорные входы и на основе ассоциативной памяти выполняет процедуру распознавания образов.

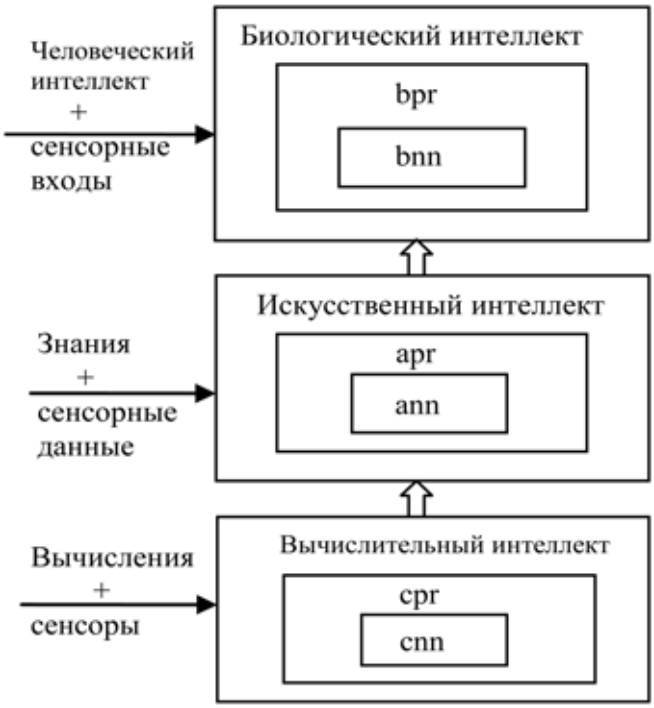

Рис. 3. Архитектура интеллектуальной системы ABC
(a — artificial, b — biological, c — computational, pr — pattern recognition, nn — neural network)

В модели Besdek J. вычислительный и искусственный интеллект являются строительными блоками для построения более сложного биологического интеллекта. Вычислительный интеллект имеет дело только с числовыми данными, обладает способностью распознавания и не использует знания в смысле ИИ, но при этом вычислительные нейроподобные системы являются моделями, которые появились из биологии. Эти компоненты могут быть нейронными сетями (НС) с прямым распространением сигналов, самоорганизующимися сетями Кохонена и т. д., представляться с помощью генетических алгоритмов или эволюционных моделей. Системы, основанные на явных знаниях или на прецедентах, являются промежуточным уровнем; сюда же можно отнести когнитивные модели, имитирующие работу отдельных элементов мозга. Модели на основе нечеткой логики аккумулируют числовую и семантическую информацию и также являются компонентами символьного уровня.

Создание гибридных моделей ГС в смысле Besdek связано с расширением возможностей низкоуровнего вычислительного интеллекта за счет высокоуровнего ИИ в целях реализации биологического интеллекта.

IRIS-модель (Integration of Reasoning, Informing and Serving)

Эта гибридная модель была предложена Soucek B. [6] и предназначалась для создания более эффективной интеллектуальной технологии для бизнеса в целях достижения основных показателей производства: темпов, качества, минимальных затрат. Модель базируется на системном подходе к созданию ИС и включает как инженерные методики, так и ряд интеллектуальных компонентов, а именно — методы, используемые в научных дисциплинах, таких как биология, когнитивная психология, лингвистика, эпистемология, компьютерные науки. Важность создания таких гибридных моделей заключается в возможности обработки разнообразных данных и обобщении знаний.

IRIS включает следующие компоненты:

- ряд интеллектуальных технологий (ЭС, НС, и др.);
- методику интеграции (правила объединения блоков);
- стандартные программные модули;
- специальные языки и трансляторы;
- программная поддержка оборудования;
- средства разработки приложений;

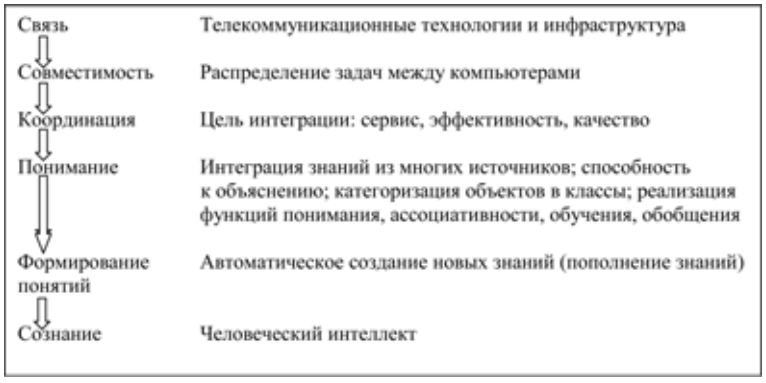

Рис. 4. Уровни интеграции в модели IRIS

- модули контроля и автоматической обработки данных и знаний;
- средства интерактивного взаимодействия с БД;
- преобразователи сигнал—символ, образ—категория, строящиеся на основе использования ЭС и НС в качестве пре- и постпроцессоров.

На рис. 4 показаны уровни интеграции компонентов, которые могут быть использованы в модели IRIS. В настоящее время наиболее высоким уровнем, реализованным в этой гибридной модели, является уровень понимания (распознавания).

Для реализации более высоких уровней требуется решение проблем организации диалога с системой на естественном языке, возможности обмена информацией любого типа (текста, графики) между уровнями, возможности представления и обработки нечеткой и неопределенной информации. В настоящее время это наиболее перспективные направления исследования принципов построения ИС.

Классификация по L. Medsker

L. Medsker [3] предложил пять типов гибридных моделей (рис. 5), классифицируемых по степени связанности отдельных модулей: автономные, трасформационные, слабо связанные, сильно связанные, полностью интегрированные.

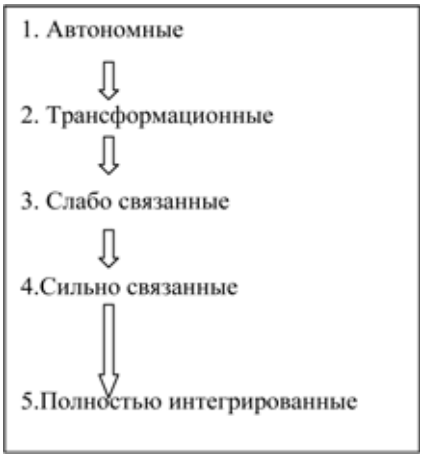

Рис. 5. Классификация гибридных моделей по L. Medsker

Автономные модели состоят из независимых программных компонентов, которые не взаимодействуют между собой при решении проблем ИИ. Использование таких моделей преследует несколько целей:

- возможность сравнения способов решения задач определенного типа для выбора наилучшего;
- использование моделей в параллельном режиме для сокращения времени получения результата;
- использование одной технологии после окончания работы другой может подтвердить или опровергнуть полученный результат;
- использование одной технологии для быстрого создания прототипа ИС, который затем будет развиваться на основе другой технологии; например, НС может быть быстро обучена на конкретных данных, а более полное исследование проблемы будет впоследствии проведено с помощью ЭС [10].

Достоинством таких моделей является простота и легкость использования коммерчески доступных программ. Недостатки автономных моделей заключаются в отсутствии возможности трансформации моделей друг в друга при решении сложных задач и необходимости одновременного обновления данных.

Трансформационные модели имеют возможность трансформации друг в друга, в отличие от автономных, при решении одной задачи в целях получения наилучшего результата. Наиболее часто используются гибридные модели ЭС $\rightarrow$ НС и НС $\rightarrow$ ЭС. Анализ данных и предварительная обработка знаний – основные области применения таких моделей. Модель ЭС $\rightarrow$ НС работает в том случае, когда с помощью ЭС невозможно получить адекватного результата или когда от ИС требуется более высокая скорость обработки данных и знаний, адаптивность и робастность. Кроме того, правила ЭС часто используются в качестве обучающей выборки для обучения НС. Модель НС $\rightarrow$ ЭС применяется для быстрой кластеризации данных, фильтрации ошибок в данных, обобщения данных и формирования базы правил.

Слабо связанные модели — это, по существу, первая реальная форма интеграции ИС. Приложение распределяется между различными интеллектуальными компонентами, которые взаимодействуют через файлы данных.

Достоинства слабо связанных систем заключаются в простоте разработки и использовании в качестве программного обеспечения коммерчески доступных программ, снижающих время на про-

граммирование. Простота реализации интерфейсов файлов данных снижает время получения решения. Недостатки этих типов гибридных моделей связаны с высокой стоимостью коммуникаций и увеличением времени на выполнение операций.

Категории слабо и сильно связанных моделей имеют значительное перекрытие. Обе модели включают независимые компоненты, например НС и ЭС, однако взаимодействие между компонентами в процессе решения задачи, в отличие от слабо связанных моделей, осуществляется через резидентную память, что значительно увеличивает производительность гибридной ИС и усиливает ее интерактивные возможности. Сильно связанные системы могут быть представлены теми же компонентами, что и слабо связанные системы: пре- и постпроцессорами, сопроцессорами, но которые работают быстрее.

Сильно связанные системы имеют низкие коммуникационные затраты и более высокую производительность по сравнению со слабо связанными. Основные недостатки: увеличение сложности ИС для реализации внутреннего интерфейса данных; необходимость сбора и обработки больших массивов данных, которые могут содержать избыточную информацию; трудность контроля и тестирования ИС, особенно встроенных систем.

Полностью интегрированные модели имеют общие структуры данных и знания. Взаимодействие между различными компонентами осуществляется на уровне методов. Наиболее известным гибридом этого класса являются нейронечеткие системы [8, 9, 13, 14], в которых, например, многослойная нейронная сеть моделирует нечеткий вывод или нечеткая нейронная сеть выполняет нечеткую кластеризацию. Преимущества полной интеграции заключаются в том, что ИС, построенные на такой основе, могут одновременно обладать такими возможностями, как адаптивность, обобщение, обучаемость, устранение шума или ослабление его влияния на конечный результат, использование логической дедукции. Эти возможности не достижимы в ИС с единственной интеллектуальной технологией.

Классификация по W. Wermter

Схемы классификации и терминология описания гибридных систем, предложенные L. Medsker и другими авторами, связаны с анализом степени взаимодействия между интеллектуальными модулями и иногда перекрываются. S. Wermter [4] поставил задачу создания схемы классификации, которая бы включила все или большую часть основных особенностей предложенных ранее классификаций, а также учитывала бы тенденцию развития новых интеллектуальных технологий, которые еще не были включены в классификационные схемы. В этом смысле S. Wermter рассматривает HIS как непрерывно развивающуюся (эволюционную) систему. Предложенная им классификационная схема включает три основные группы.

Первая группа — это унифицированные гибридные системы, состоящие из специализированных нейросетевых компонентов, выполняющих функции символьных систем. Унифицированные системы строятся по двухуровневому способу: нижний уровень состоит из многих НС, каждая из которых реализует одно из правил базы правил, а верхний уровень — это НС, которая из многих правил в реальном масштабе времени выбирает одно активное правило. Несмотря на эффективность работы таких систем, они имеют ограниченное применение вследствие сложности разработки.

Вторая группа — трансформационные модели, использующие два формата представления: в виде ЭС и НС, которые могут преобразовываться в процессе функционирования HIS. Наиболее интересные особенности таких систем — возможность вставки, извлечения и обновления знаний.

Третья группа — модульные системы, функционирующие подобно биологическим системам, встречающимся в природе. Как правило, такие системы включают подсистемы, ответственные за реализацию определенных функций. Модульные ГС содержат несколько НС и несколько баз правил, которые могут иметь различную степень интеграции, при этом в процессе функционирования структура модулей не меняется. Поэтому для работы такой системы целесообразно создавать нейросетевую базу знаний, которая будет содержать набор модулей (обученных НС) для решения определенных проблем. Главная особенность модульных ГС — возможность параллельной обработки информации для повышения скорости или надежности вычислений.

Классификация по Kasabov N.

Дальнейшее направление исследований в области построения гибридных систем связывается с созданием эволюционных, постоянно развивающихся систем, работающих в режиме on-line и подстраивающихся под конкретную решаемую задачу. Для этого должны быть предложены новые модели машинного интеллекта, включающие различные типы НС, созданные по модульному и иерархическому принципам, новые нейронечеткие и адаптивные системы (рис. 6). Kasabov N. в [1] предложил проект многоуровневой постоянно развивающейся эволюционной ГС, на каждом уровне которой возможна подстройка к решаемой задаче:

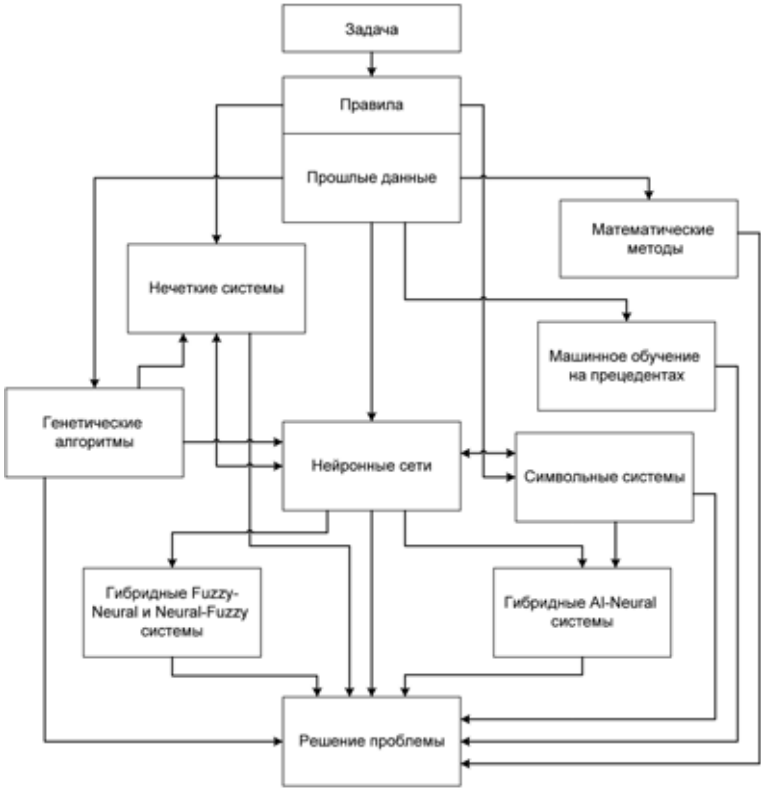

Рис. 6. Способы объединения интеллектуальных технологий в ГС

- **генетический уровень** определяет вид и числовые параметры активационной функции НС, порог (bias), способ агрегации входов, число входов и т. д.;
- **нейронный уровень** реализует определенную функцию в системе и определяет радиус рецептивной области взаимодействия с другими нейронами;
- **ансамбль нейронов** характеризуется структурой, способом распространения сигналов по НС, числом слоев и нейронов в каждом слое, алгоритмом обучения, функцией оценки и интерпретации результатов работы сети;
- **уровень многомодульной иерархической структуры** определяет число и тип нейросетевых модулей системы, механизмы взаимодействия между отдельными модулями и принципы обмена информацией с НЛ для объяснения результатов решений, способ организации блока принятия решений, связь с внешней средой;
- **популяция эволюционных систем** обеспечивает их развитие и взаимодействие на основе использования генетических алгоритмов.

Заключение

Основная цель исследования принципов построения эволюционных ГС заключается в модификации известных методов и технологий ИИ для обработки разнотипной информации в реальном масштабе времени. Другая, не менее важная цель — исследование принципов функционирования мозга и генетики для создания новых вычислительных моделей ГС в практических приложениях.

Список литературы

1. **Kasabov N.** Evolving connectionist systems. Springer-Verlag London Limited. 2003.
2. **Jang J.-S. R., Sun C.-T., Mizutani E.** Neuro-Fuzzy and Soft computing. Prentice-Hall, Inc. 1997.
3. **Medsker L.** Hybrid intelligent systems. Kluver Academic Publishers. 1998.
4. **Wermter S.** Hybrid approaches to neural network-based language processing // Technical Report TR-97-030. International Computer Science Institute, Berkely, California, 1997.
5. **Bezdek J.** What is computational intelligence?, in [Zurada J., Marks I., Robinson C. Computational Intelligence: Imitating Life] // IEEE Press, New York, 1994.
6. **Soucek B.** and the IRIS Group (eds). Neural and Intelligent Systems Integration. New York: John Wiley and Sons, 1991.
7. **Колесников А. В.** Гибридные интеллектуальные системы. Теория и технология разработки. СПб.: Изд-во СПбГТУ, 2001.
8. **Лю Б.** Теория и практика неопределенного программирования: Пер. с англ. / Под ред. Ю. В. Тюменцева. М.: Бином. Лаборатория знаний. 2005.
9. **Комарцова Л. Г., Максимов А. В.** Нейрокомпьютеры: Учеб. пособие. М.: МГТУ им. Н. Э. Баумана. 2004.
10. **Комарцова Л. Г.** Интеллектуальные системы. Проблемы создания эволюционных коннекционистских систем. М.: Физматлит. 2005. С. 39—47.
11. **Новак В., Перфильева И., Мочкорж И.** Математические принципы нечеткой логики: Пер. с англ. / Под ред. А. Н. Аверкина. М.: Физматлит. 2006.
12. **Вагин В. Н., Головина Е. Ю., Загорянская А. А., Фомина М. В.** Достоверный и правдоподобный вывод в интеллектуальных системах. 2-е изд. М.: Физматлит. 2008.
13. **Рутковская Д., Пилиньский М., Рутковский Л.** Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И. Д. Рудинского. М.: Горячая линия. Телеком, 2004.
14. **Нечеткие** гибридные системы. Теория и практика / Под ред. Н. Г. Ярушкиной. М.: Физматлит. 2007.
15. **Гладков Л. А., Курейчик В. В., Курейчик В. М., Сороколетов П. В.** Биоинспирированные методы в оптимизации. М.: Физматлит. 2009.
16. **Кохонен Т.** Самоорганизующиеся карты: Пер. с англ. / Под ред. Ю. В. Тюменцева. М.: Бином. Лаборатория знаний. 2008.
17. **Емельянов В. В., Курейчик В. В., Курейчик В. М.** Теория и практика эволюционного моделирования. М.: Физматлит. 2003.
18. **Ярушкина Н. Г.** Основы теории нечетких и гибридных систем: Учеб. пособие. М.: Финансы и статистика. 2004.
19. **Herrera F., Lozano M., Verdegay J.** The use of fuzzy connectives to design real-coded ge-netic algorithms // Mathware and Soft computing. 1995. N 1.
20. **Herrera F., Verdegay J.** (eds) Genetic algorithms and soft computing. Heidelberg. Physica-Verlag, 1996.
21. **Норенков И. П.** Исследование эффективности генетического метода с фрагментным кроссовером // Информационные технологии. 2008. № 6. С. 26—29.
22. **Back T.** Evolutionary algorithms in theory and practice. NY: Oxford Unipress. 1996.
23. **Родзин С. И.** Интеллектуальные системы. Проблемы и перспективы создания единой концепции гибридных эволюционных вычислений. М.: Физматлит. 2005. С. 76—94.
24. **Оптимизация** на основе методов гомеостатики, эволюционного развития и самоорганизации. / Под ред. В. М. Курейчика. Таганрог: Изд-во ТРТУ. 2006.

УДК 004.8 + 544.4

В. Е. Туманов, канд. хим. наук,
зав. сектором информационного обеспечения научных исследований,
Институт проблем химической физики РАН,
г. Черноголовка,
e-mail: tve90@yandex.ru

Применение искусственной нейронной сети для прогноза реакционной способности молекул в радикальных реакциях

Рассмотрено применение искусственной нейронной сети прямого распространения, обученной по экспериментальной выборке, для оценки реакционной способности органических молекул в радикальных реакциях. Приведены результаты обучения и предсказания нейросети. Рассмотрена реализация нейросети как веб-сервиса предметно-ориентированной системы научной осведомленности по физической химии радикальных реакций.

Ключевые слова: *искусственные нейронные сети, предметно-ориентированная система научной осведомленности, экспертная система, веб-сервисы, реакционная способность органических молекул*

Введение

В настоящее время широкое распространение в решении прикладных задач автоматизированной обработки научных данных получили искусственные нейронные сети (ИНС). Основные направления применения ИНС в химических и биохимических исследованиях рассмотрены в обзоре [1]. Большинство работ в этой области посвящено корреляции между строением химических соединений и проявляемыми ими физико-химическими свойствами или биологической активностью. В физической химии основными направлениями применения ИНС является моделирование химических процессов, моделирование динамических свойств молекул и систем.

С одной стороны, в физической химии радикальных реакций накоплены значительные массивы экспериментальных данных по реакционной способности (констант скорости или энергии активации) молекул в радикальных реакциях [2, 3]. С другой стороны, постановка экспериментов по количественному определению реакционной способности молекул в радикальных реакциях является дорогостоящей и трудоемкой задачей. Проведение квантово-химических расчетов занимает много времени, а полученные в результате таких расчетов данные не обладают достаточной надежностью. Поэтому разработка ИНС на основе уже имеющихся экспериментальных данных для прогнозирования реакционной способности органических молекул в радикальных реакциях представляется актуальной задачей.

Знание о реакционной способности органических молекул в радикальных реакциях необходимо при разработке новых органических материалов, конструировании новых лекарственных препаратов, проектировании технологических процессов, планировании и проведении научного эксперимента, профессиональной подготовки студентов и аспирантов.

В настоящей статье рассматривается применение ИНС прямого распространения для оценки реакционной способности органических молекул в радикальных бимолекулярных реакциях в жидкой фазе и представление ИНС в виде веб-сервиса для использования в системе научной осведомленности по физической химии в сети Интернет [4].

Постановка задачи

Экспериментально реакционная способность органической молекулы в радикальной реакции определяется энергией активации E или классическим потенциальным барьером E_e:

$$E_e = E - 0{,}5(hL\nu_i - RT), \qquad (1)$$

где h — постоянная Планка; L — число Авогадро; ν_i — частота колебания разрываемой связи; R — газовая постоянная; T — температура.

Константа скорости k химической реакции вычисляется по формуле

$$k = nA_0\exp(-E/RT), \qquad (2)$$

где A_0 — частота столкновений на одну реакционную связь; n — число эквиреакционноспособных связей в молекуле.

При конструировании информационного пространства для ИНС предсказания реакционной способности используется функциональная связь между реакционной способностью химической реакции и термохимическими характеристиками молекулы (энтальпией реакции ΔH).

Впервые на функциональную связь между реакционной способностью и энтальпией реакции обратил в начале прошлого века академик Н. Н. Семенов (известное соотношение Поляни—Семенова [5]):

$$E = B - \gamma\Delta H, \qquad (3)$$

где B и γ — эмпирические коэффициенты.

В работах [6, 7] были предложены эмпирические модели элементарной радикальной реакции,

которые позволили построить нелинейные корреляционные зависимости между классическим потенциальным барьером реакции и ее термохимическими характеристиками:

- аппроксимация указанной выше зависимости по работе [6] параболой

$$br_e = \alpha\sqrt{E_e - \Delta H_e} - \sqrt{E_e}; \qquad (4)$$

- аппроксимация неявно заданной кривой по работе [7]

$$br_e = D_{ei}^{1/2}\ln\left(\frac{D_{ei}^{1/2}}{D_{ei}^{1/2} - E_e^{1/2}}\right) + \alpha D_{ef}^{1/2}\ln\left(\frac{D_{ef}^{1/2}}{D_{ef}^{1/2} - (E_e - \Delta H_t)^{1/2}}\right). \qquad (5)$$

Параметры, входящие в формулы (4)—(5), описаны ниже.

Согласно предложенным эмпирическим моделям в предположении о гармоничности валентных колебаний реакция радикального отрыва $R^o + R_1H \rightarrow RH + R_1^o$ (где R^o и R_1^o — алкильные радикалы, а RH и R_1H — углеводородные молекулы) характеризуется следующими параметрами:

- энтальпией $\Delta H_e = D_i - D_f + 0{,}5(hL\nu_i - hL\nu_f)$, включающей разницу энергий нулевых колебаний рвущейся и образующейся связей (представляет собой изменение потенциальной энергии системы), где ν_i — частота колебания молекулы вдоль разрываемой связи; ν_f — частота колебания молекулы вдоль образующейся связи; D_i — энергия диссоциации разрываемой связи; $D_{ei} = D_i + 0{,}5hL\nu_i$; D_f — энергия диссоциации образующейся связи; $D_{ef} = D_f + 0{,}5hL\nu_f$;
- классическим потенциальным барьером активации E_e (1), который включает в себя энергию нулевого колебания рвущейся связи;
- параметром r_e, который равен суммарному растяжению рвущейся и образующейся связей в переходном состоянии;
- параметрами $b = \pi(2\mu_i)^{1/2}\nu_i$ и $b_f = \pi(2\mu_f)^{1/2}\nu_f$, которые описывают зависимость потенциальной энергии от амплитуды колебаний атомов вдоль рвущейся (i) и образующейся (f) валентных связей ($2b^2$ — силовая постоянная связи, μ_i — приведенная масса атомов для разрываемой связи, μ_f — приведенная масса атомов для образующейся связи);
- параметром α (α^2 равен отношению силовых постоянных рвущейся и образующейся связей);
- предэкспоненциальным множителем A_0 в расчете на одну эквиреакционную связь в молекуле.

По статистически определенному значению параметра br_e исходя из формулы (4) можно оценить значение классического потенциального барьера по формуле

$$\sqrt{E_e} = \frac{br_e}{1-\alpha^2}\left(1 - \alpha\sqrt{1 - (1-\alpha^2)\frac{\Delta H_e}{(br_e)^2}}\right). \qquad (6)$$

Таким образом, можно предположить, что зависимость классического потенциального барьера E_e от термохимических характеристик реагентов и кинетических характеристик радикальной реакции можно представить в виде функции

$$E_e = F(\Delta H_e,\ T,\ nA_0,\ \alpha). \qquad (7)$$

Тогда задача работы ИНС по предсказанию значения энергии активации E_e как функции от термохимических и кинетических характеристик реагентов с последующим вычислением константы скорости реакции по формулам (1) и (2) сводится к аппроксимации неизвестной функции (7).

Искусственная нейронная сеть для прогнозирования реакционной способности молекул

Для аппроксимации зависимости (7) была использована ИНС прямого распространения [8] с типовой архитектурой, приведенной на рис. 1. Используемая нами ИНС имеет четыре входа, три внутренних слоя (каждый из семи нейронов) и один выход.

Работа ИНС задается формулами

$$\begin{gathered} NET_{jl} = \sum_i w_{ijl}x_{ijl}; \\ OUT_{il} = \Phi(NET_{jl} - \theta_{jl}); \\ x_{ij(l+1)} = OUT_{il}, \\ \delta = 0{,}5\sum_j\sum_k (y_j^k - d_j^k)^2, \end{gathered} \qquad (8)$$

где индексом i всегда будем обозначать номер входа; j — номер нейрона в слое; l — номер слоя; x_{ijl} — i-й входной сигнал j-го нейрона в слое l; w_{ijl} — весовой коэффициент i-го входа j-го нейро-

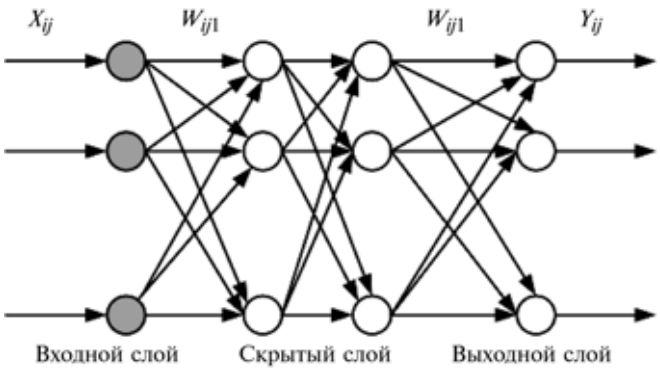

Рис. 1. Типовая архитектура искусственной нейронной сети прямого распространения

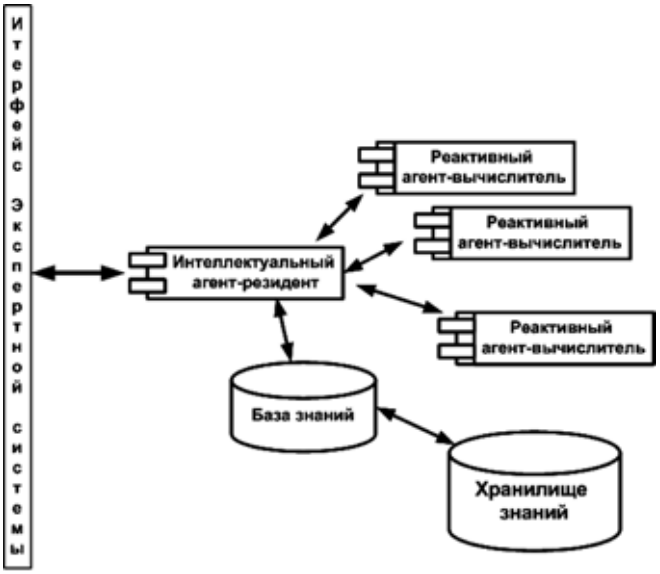

Рис. 2. Использование построенной искусственной нейронной сети в системе научной осведомленности по физической химии радикальных реакций

Результаты обучения ИНС

Реакция	E	E_e	$E_{\text{ИНС}}$
$CH_3 + CH_2ClBr$	27,17	36,57	30,97
$C_6H_5 + (CH_3)_4C$	23,82	43,17	19,00
$CCl_3 + CH_3(CH_2)_5CH_3$	46,88	67,35	52,35
CH_3 + цикло-$[(CH_2)_6]$	44,17	46,80	42,13
C_6H_5 + цикло-$[(CH_2)_5]$	27,99	31,04	27,48
CH_3 + цикло-$[CH(CH_3)(CH_2)_4]$	30,23	39,67	33,18
$CCl_3 + C_6H_5CH_3$	44,30	52,62	40,88
$C_6H_5 + C_6H_5CH_3$	20,67	24,45	16,01

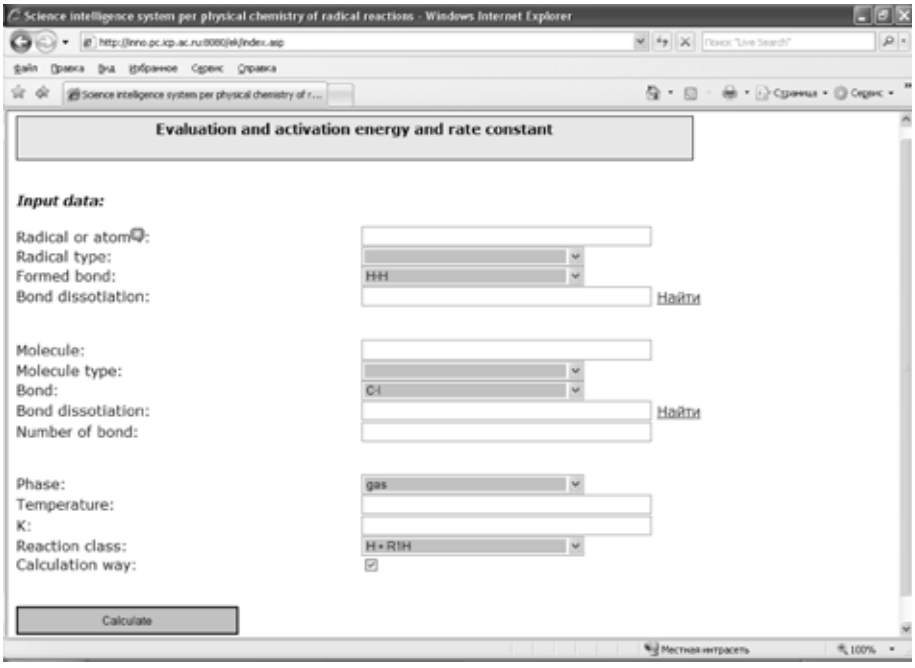

Рис. 3. Установка признака подключения искусственной нейронной сети на интерфейсе экспертной системы прогноза реакционной способности молекул в радикальных реакциях

на в слое l; NET_{jl} — сигнал NET j-го нейрона в слое l; OUT_{jl} — выходной сигнал нейрона; θ_{jl} — пороговый уровень j-го нейрона в слое l; x_{ji} — входной вектор-столбец слоя l.

Входной вектор ИНС задается в виде вектора $x_0 = \{T,\ D_{ei},\ D_{ef},\ nA_0,\ k,\ \alpha\}$, выходное значение равно E_e.

В качестве алгоритма обучения был использован метод обратного распространения ошибки [8]. Функция активации является сигмоидальной функцией и задается формулой

$$f(x) = \frac{1}{1 + \mathbf{e}^{-\beta x}}, \qquad (9)$$

где параметр $\beta > 0$ подбирается экспериментально.

Результаты обучения и предсказания искусственной нейронной сети

Для обучения ИНС потребовалось 3000 итераций на обучающей выборке из 295 образцов. Обучающая выборка была построена из элементарных радикальных реакций R′ + RH в жидкой фазе, где R′ — радикал, а RH — углеводородная молекула.

В таблице приведено сравнение предсказания значений классического потенциального барьера реакции с использованием ИНС ($E_{\text{ИНС}}$), экспериментальных значений (E) и вычисленных по формуле (6) значений классического потенциального барьера (E_e).

Погрешность предсказания значений классического потенциального барьера радикальной реакции с помощью ИНС на контрольной выборке (из 20 образцов) составляет 3,34 ± ± 2,0 кДж/моль, что находится в пределах погрешности эксперимента (± 4 кДж/моль). Погрешность предсказания значений классического потенциального барьера радикальной реакции по формуле (6) на той же контрольной выборке составляет 9,5 ± ± 7,0 кДж/моль. ИНС делает предсказание лучше, чем рассчитанное по формуле (6). Это обусловлено величиной статистической ошибки br_e, которая определяет класс радикальных реакций. Таким образом, ИНС лучше аппроксимирует функциональную зависимость (7) за счет вычисления весовой матрицы связей.

Практическое использование искусственной нейронной сети для предсказания реакционной способности реагентов в радикальных реакциях

В настоящее время веб-сервисы стали распространенным средством программного обеспечения промежуточного уровня для выполнения различных операций и процедур в рамках портальных технологий. Разработанная ИНС для предсказания реакционной способности реагентов предназначена для использования в предметно-ориентированной системе научной осведомленности по физической химии радикальных реакций [4] в целях пополнения ее хранилища знаний. ИНС реализована как веб-сервис экспертной системы для оценки реакционной способности реагентов в радикальных реакциях [9]. Механизм использования ИНС приведен на рис. 2.

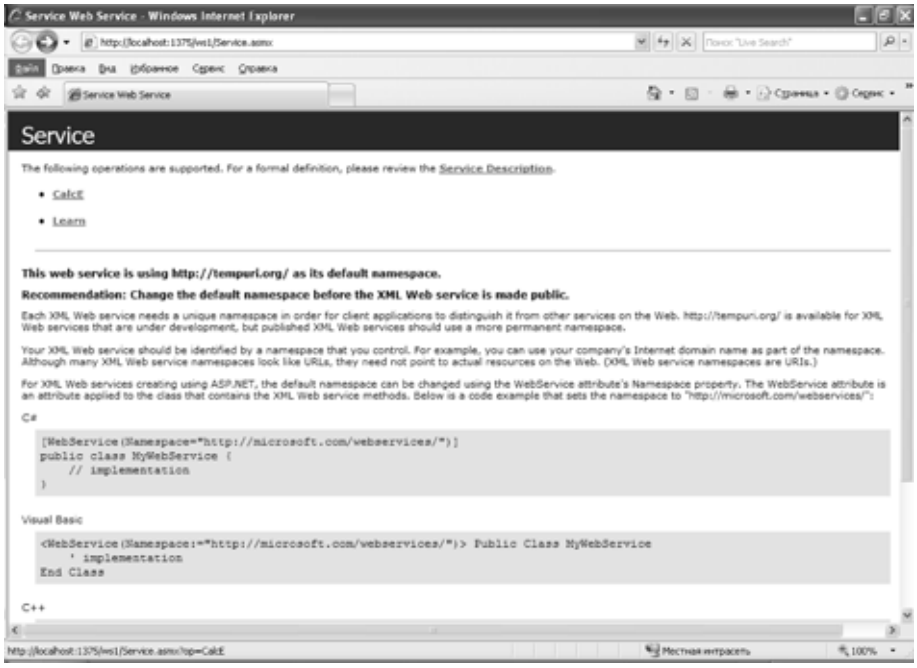

Рис. 4. Интерфейс искусственной нейронной сети в автономном режиме

В нашем случае веб-приложение (экспертная система) посылает веб-сервису вектор исходных данных. Веб-сервис выполняет следующие задачи:

- принимает от веб-приложения входные данные;
- преобразует их;
- анализирует условия применимости ИНС;
- активизирует ИНС;
- получает результат работы ИНС;
- формирует выходные данные;
- возвращает их веб-приложению.

Код для взаимодействия веб-приложения с ИНС приведен ниже:

```
var SOAPRequest = new ActiveXObject("MSXML. DOMDocu-
ment");
SOAPRequest.async = false;
var SOAPResponse = new ActiveXObject("MSXML. DOMDocu-
ment");
SOAPResponse.async = false;
var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
xmlHttp.Open("POST",
"http://192.168.2.199/ws1/service.asmx/CalcE",
false);
xmlHttp.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");
xmlHttp.send("T = "+ti+"&n = "+nod+"&A = "+r_class.
Fields.Item("AL").Value+
"&Di = "+di+"&Df = "+df+"&alpha = "+r_class.Fields.
Item("ALPHA").Value);
SOAPResponse.loadXML(xmlHttp.responseXML.xml);
var xmldoc = xmlHttp.responseXML;
var root_node = xmldoc.getElementsByTag-
Name('float').item(0);
var WSE;
WSE = root_node.firstChild.data;
```

На рис. 3 показан интерфейс экспертной системы, на котором указывается необходимость использования ИНС, а на рис. 4 — интерфейс обучения и отладки ИНС в автономном режиме.

Заключение

Впервые для аппроксимации по экспериментальным данным функциональной зависимости классического потенциального барьера химической реакции от термохимических характеристик реагентов и кинетических характеристик реакции была использована ИНС прямого распространения.

Результаты предсказания реакционной способности для жидкофазных реакций углеводородов с углеводородными радикалами находятся в пределах погрешности эксперимента.

Реализация ИНС в виде веб-сервиса и публикация ее в Интернет в рамках системы научной осведомленности позволяет получать оценки скорости радикальных реакций в жидкой и газовой фазах широкому кругу заинтересованных пользователей.

Список литературы

1. **Баскин И. И., Палюнин В. А., Зефиров Н. С.** Применение искусственных нейронных сетей в химических и биохимических исследованиях // Вестник МГУ им. М. В. Ломоносова. Сер. Химия. 1999. Т. 40. № 5. С. 323—326.

2. **Денисов Е. Т., Туманов В. Е., Денисова Т. Г., Дроздова Т. И., Покидова Т. С.** Реализация Банка кинетических констант радикальных жидкофазных реакций на IBM PC/AT. Черноголовка: ИХФЧ РАН. Препринт. 1992. 58 с.

3. **Mallard W. G., Westley F., Herron J. T., Hampson R. F.** NIST Chemical Kinetics Database — Ver. 6.0. NIST Standard Reference Data. 1994.
4. **Туманов В. Е.** Предметно-ориентированные системы научной осведомленности // Информационные технологии. 2009. № 5. С. 12—18.
5. **Семенов Н. Н.** О некоторых проблемах химической кинетики и реакционной способности. М.: Изд. АН СССР. 1958. 686 с.
6. **Денисов Е. Т.** Нелинейные корреляции в кинетике радикальных реакций. Препринт ОИХФ АН СССР. Черноголовка. 1990. 18 с.
7. **Денисов Е. Т., Туманов В. Е.** Модель переходного состояния как результат пересечения двух термов Морзе в приложении к реакциям атомарного водорода // Журнал физической химии. 1994. Т. 68. № 4. С. 719—725.
8. **Гладышев И.** Анализ и обработка данных: специальный справочник. СПб.: Питер, 2001. 752 с.
9. **Лазарев Д. Ю., Прохоров А. И., Туманов В. Е.** Экспертная система в Интернет для оценки реакционной способности молекул в радикальных реакциях // Приложение к журналу "Открытое образование". Матер. XXXIV Междунар. конф. и дискуссионного научного клуба "Информационные технологии в науке, социологии, экономике и бизнесе", IT + SE'08, Осенняя сессия. Украина, Крым, Ялта—Гурзуф, 30 сентября — 8 октября 2008 г. С. 63—65.

УДК 004.056.53

В. И. Глова, д-р техн. наук, проф.,
e-mail: glova@kai.ru,
А. С. Катасёв, канд. техн. наук, доц.,
e-mail: Kat_726@mail.ru,
Г. С. Корнилов, аспирант,
e-mail: kegork@mail.ru,
Казанский государственный технический университет им. А. Н. Туполева

Преднастройка и оптимизация параметров нечеткой нейронной сети при формировании баз знаний экспертных систем

Для повышения точности аппроксимации экспериментальных данных предлагается методика преднастройки и оптимизации параметров нечеткой нейронной сети. Проведена оценка эффективности методики путем реализации ее методов и алгоритмов в системе моделирования MatLab и апробация ее работы при обучении нейронной сети на статистических данных электронной почты и медицинской диагностики.

Ключевые слова: *интеллектуальный анализ данных, нечеткие нейронные сети, экспертные системы, принятие решений*

Введение

Современные информационные системы анализа огромных массивов информации или управления сложными процессами невозможно представить без элементов искусственного интеллекта, реализованного, как правило, в экспертных системах (ЭС), моделирующих процесс рассуждения эксперта при принятии им решения. Основным элементом ЭС является база знаний, представленная множеством правил, описывающих закономерности в предметной области. Поэтому проектирование баз знаний является важнейшей задачей при разработке экспертных систем.

Традиционные подходы получения знаний, включающие процессы их извлечения и приобретения, имеют ряд ограничений [3], актуализирующих необходимость разработки методов, алгоритмов и реализующих их программных комплексов автоматизированного формирования баз знаний ЭС. В решении данной задачи распространение получили такие направления искусственного интеллекта, как нейронные сети и нечеткие системы. Значительную актуальность приобрело создание нечетких нейронных сетей (ННС), сочетающих в себе достоинства обоих направлений [3].

Постановка задачи

Проведенный анализ возможностей нечетких нейронных сетей показал, что качество их обучения в значительной степени зависит от выбора числа нечетких гранул для входных лингвистических переменных (ЛП). Введение слишком большого числа значений ЛП приводит к затруднению человеком выбора одного из них в некоторой ситуации. Вместе с тем, необоснованное уменьшение числа значений ЛП приводит к недостаточности информации для человека при описании некоторой ситуации.

Кроме того, для описания закономерностей в предметной области нечеткими правилами продукций необходимы процедуры выбора оптимальных форм и параметров соответствующих функций принадлежности (ФП). Использование той или иной ФП часто определяется спецификой решаемой задачи или квалификацией эксперта. При этом от того, насколько точно выбранная функция отражает знания эксперта, во многом зависит адекватность нечетких моделей. Критерием адекватности может служить "естественность" заключений, получаемых на основе этих моделей. Подобный критерий является обобщением известного теста Тьюринга [1].

Цель данной работы — повышение точности аппроксимации экспериментальных данных нечеткими продукциями путем автоматического выбора оптимального числа значений входных лингвистических переменных и форм их функций принадлежности. Для достижения поставленной цели необходима разработка методики преднастройки и оптимизации параметров нечеткой нейронной сети.

Разработка методики преднастройки и оптимизации параметров нечеткой нейронной сети

Часто проблему выбора числа нечетких градаций решают эксперты. Из чисто психологических соображений они выбирают нечетное число значений ЛП, например, 3, 5, 7. При этом данный выбор происходит субъективно и не всегда отражает реальную картину в распределении значений лингвистической переменной. При использовании автоматических методов выбор числа значений ЛП осуществляется на основе критерия оптимальности. Для этого часто используют методы кластерного анализа.

Традиционные алгоритмы кластеризации основаны на допущениях, которые определяют основные факторы, не позволяющие применить их в разрабатываемой методике. Во-первых, являются неприемлемыми априорные предположения о свойствах кластеров, принципах объединения объектов или задание числа кластеров. Во-вторых, является неприемлемым построение алгоритма лишь на отношении точек к центрам кластеров, а не на основе взаимного расположения точек. И, наконец, недопустимо отсутствие понятной лингвистической интерпретации разбиений.

В основу разработанного алгоритма кластеризации положен аппарат нечетких отношений, использующий понятие отношения α-толерантности и α-квазиэквивалентности, имеющие, соответственно, смысл попарного сравнения образцов данных относительно заданного образца и межгруппового сравнения данных [2]. Пусть значения x_i входных параметров ННС заданы на непустом множестве X. Введем ряд понятий и определений.

Нормальной мерой сходства по расстоянию точки данных x_i с точкой x_q называют такую меру, которая достигает своих граничных значений на множестве X с функцией принадлежности, определяемой по формуле

$$\mu_{x_q}(x_i) = 1 - \frac{d(x_q, x_i)}{\max\limits_{k \in [1, Q]}(d(x_q, x_i))}, \qquad (1)$$

где $d(x_q, x_i) = \sum\limits_{i=1}^{Q} |x_q - x_i|$; Q — число значений входной переменной.

На основании данной меры сходства построим семейство α-толерантных отношений для определения степени схожести двух точек данных относительно некоторой заданной точки.

Относительной мерой сходства двух точек данных относительно точки x_0 называют функцию $\xi_{x_0} : X^2 \to [0, 1]$, $x_0 \in X$, определяемую как

$$\xi_{x_0}(x_1, x_2) = 1 - |\mu_{x_0}(x_1) - \mu_{x_0}(x_2)|, \qquad (2)$$

где $\mu_{x_0}(x)$ — нормальная мера сходства.

Мерой сходства точек данных на множестве X называют функцию ξ: $X^2 \to [0, 1]$, которая определяется как

$$\xi(a, b) = T(\xi_{x_i}(a, b), ..., \xi_{x_Q}(a, b)), \qquad (3)$$

где T — t-норма (например, min), $\xi_{x_i}(a, b)$ — относительная мера сходства, $x_i \in X$, $i = \overline{1, Q}$, $a, b \in X$. Полученное отношение является отношением α-толерантности на множестве X.

Транзитивным замыканием нечеткого отношения R, определенного на множестве X, будем называть следующее нечеткое отношение:

$$\hat{R} = \bigcup_{i=1}^{|X|} R'. \qquad (4)$$

Разработанный алгоритм нечеткой кластеризации состоит из следующей последовательности шагов.

1. Вычисление расстояний между значениями входных параметров:

$$d(x_q, x_i) = \sum_{i=1}^{Q} |x_q - x_i|.$$

2. Построение для каждой точки данных $x_i = (x_{i_1}, ..., x_{i_n})$ нормальной меры сходства по формуле (1).

3. Построение относительно каждой точки данных на основании нормальной меры сходства относительной меры сходства по формуле (2).

4. Построение меры сходства точек данных по формуле (3). Полученное отношение является отношением α-толерантности на множестве X.

5. Построение по формуле (4) транзитивного замыкания отношений мер сходства точек данных на множестве X. Построенные отношения есть отношения α-квазиэквивалентности.

6. Определение оптимального числа классов эквивалентности и соответственно разбиения множества X по кластерам, используя следующий за пороговым уровень α-квазиэквивалентности.

Рассмотрим задачу выбора оптимальных форм ФП значений лингвистических переменных. Функции принадлежности нечеткого множества тради-

ционно строят по экспертной информации. Существует большое число таких методов, которые делят на прямые и косвенные.

Примерами прямых методов являются непосредственное задание ФП таблицей, графиком или формулой. В косвенных методах значения ФП выбирают таким образом, чтобы удовлетворить заранее сформулированным критериям оптимальности. Недостатком обеих групп методов является большая доля субъективизма.

Иной подход к построению ФП основывается на параметрической идентификации нечетких моделей по экспериментальным данным "входы — выход". При этом оптимизируют параметры ФП в целях минимизации отклонения между экспериментальными данными и результатами их анализа. Использование такого подхода снимает субъективизм построения ФП, однако взамен требует представительной обучающей выборки.

В данной работе предлагается новый подход к определению оптимальной формы и начальных параметров функций принадлежности по анализу распределения экспериментальных данных. В основу подхода положен алгоритм нечеткой кластеризации, формирующий оптимальное число кластеров. Поскольку кластеризуемый объект представляет собой только один признак, можно поставить в соответствие нечеткому множеству нечеткий кластер. При этом функция принадлежности нечеткого кластера будет соответствовать искомой функции принадлежности нечеткого множества.

Пусть известны следующие числовые значения некоторого показателя $(y_1, y_2, ..., y_V)$. Рассмотрим задачу построения ФП по этим данным — синтез одного нечеткого множества $\tilde{y}$, ФП которого соответствует распределению данных $(y_1, y_2, ..., y_V)$. С математической точки зрения эта задача соответствует отображению вида $(y_1, y_2, ..., y_v) \rightarrow$ $\rightarrow (\mu_1, \mu_2, ..., \mu_v)$, где μ_v — степень принадлежности элемента нечеткому множеству $\tilde{y}$, $v = \overline{1, V}$.

Тогда, зная разбиение значений рассматриваемого показателя на кластеры, центр j-го кластера $C_j = \{x_1^j, x_2^j, x_i^j, ..., x_n^i\}$ можно определить как

$$x_{\text{ц}}^j = x_k^j \Big| \sum_{i=1}^{n} d(x_k^j, x_i^j) \rightarrow \min_k . \qquad (5)$$

Рис. 1 иллюстрирует правило определения центра j-го кластера.

Зная центр кластера, можно определить степени принадлежности каждой точки данному кластеру по формуле

$$\mu_{C_j}(x_k^j) = 1 - \frac{d(x_k^j, x_{\text{ц}}^j)}{\sum_{i=1}^{V} d(x_i^j, x_{\text{ц}}^j)} . \qquad (6)$$

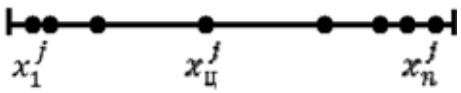

Рис. 1. Центр j-го кластера

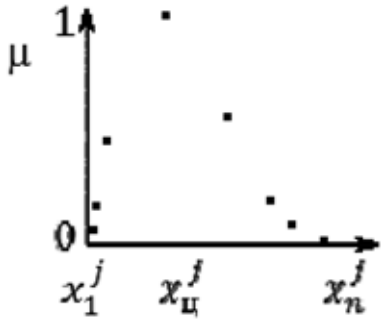

Рис. 2. Степени принадлежности точек кластеру

На рис. 2 приведен результат расчета по формуле (6).

Найденное нечеткое множество $\tilde{y}$ можно аппроксимировать подходящей параметрической функцией принадлежности, например, треугольной, трапецеидальной, гауссовой и другими. Оценка точности аппроксимации определяется значением остаточной дисперсии или ее отношением к исходной дисперсии. На практике строят несколько аппроксимаций и принимают ту, которая дает минимум указанной оценки.

Будем считать, что изучаемые процессы подчиняются некоторому закону распределения, описываемому математической моделью, и отклонения от него являются случайными. В этом случае метод наименьших квадратов является наилучшим для оценки точности аппроксимации [4]. Используя данный метод и проводя оценку значений остаточной дисперсии, определяем форму и проводим инициализацию параметров ФП, наиболее точно описывающую исходное нечеткое множество.

Алгоритм выбора оптимальных форм и начальных параметров функций принадлежности включает следующие шаги.

1. Сформировать оптимальное число значений входных параметров ННС, используя разработанный алгоритм нечеткой кластеризации.

2. Вычислить центры полученных кластеров по формуле (5).

3. Рассчитать степени принадлежности по формуле (6).

4. Провести аппроксимацию степеней принадлежности различными формами функций принадлежности.

5. Провести оценку значений остаточной дисперсии методом наименьших квадратов и выбрать оптимальную форму ФП.

Таким образом, методика преднастройки и оптимизации параметров нечеткой нейронной сети заключается в последовательном применении предложенных алгоритмов, позволяющих до начала ее работы автоматически выбрать объективно оптимальное число нечетких градаций значений входных нейронов и наилучшим образом описывающую их форму и начальные параметры функций принадлежности.

Оценка эффективности методики

В целях оценки эффективности предложенной методики реализован программный комплекс на базе разработанных алгоритмов и проведены численно-параметрические исследования в среде MatLab 7 (рис. 3).

Проведен анализ устойчивости алгоритма нечеткой кластеризации к случайным "выбросам", нетипичным и пропущенным значениям, а также анализ оптимальности его кластерного решения. Для этого исходная выборка случайным образом делилась на две примерно равные части, проводилась кластеризация обеих частей и затем посредством визуализованных кластерных решений оценивались полученные результаты.

Использовался и другой подход проверки устойчивости путем многократного размножения (дублирования) исходной выборки из N объектов, объединение всех дублированных выборок в одну большую выборку (псевдогенеральную совокупность) и случайное извлечение из нее новой выборки из N объектов. После этого проводилась кластеризация этой выборки и сравнивались полученные результаты.

Наряду с описанными подходами проводился анализ результатов, полученных алгоритмом кластеризации нечетких k-средних, в сравнении с которым разработанный алгоритм наглядно, посредством визуализации кластерных решений в графическом интерфейсе, показал устойчивость и оптимальность всех его кластерных решений.

Рис. 3. Иллюстрация работы алгоритма нечеткой кластеризации

Апробация методики

В целях апробации разработанная методика была реализована в системе предварительного выявления несанкционированной массовой рассылки (спама) на базе программного комплекса "Нечеткая нейронная сеть" [3]. Система проводит классификацию входящих электронных писем и информирует пользователя о принадлежности письма к спаму, принимая решение на основании сформированной базы знаний.

Для ее формирования наряду с экспертами использовали нечеткую нейронную сеть в комплексе с разработанной методикой преднастройки и оптимизации ее параметров, с помощью которых обрабатывалась информация, представлявшая собой значения признаков пришедшего письма.

Для эксперимента было использовано пространство признаков оформления и стиля писем, форматирования и заголовков. Обучающие выборки включали в себя поля "количество адресатов", "количество замен кириллических букв на латиницу", "частота встречаемости слов верхнего регистра", "частота встречаемости слов с пробелами". Для реализации почтового трафика при обучении ННС была использована база из 500 писем, большая часть которой содержала характерные признаки спама. Обучение проводилось с использованием треугольных функций принадлежности, установленных с применением разработанной методики.

В результате сформировано множество правил, из которых отобраны наиболее значимые, определяющие условия определения спама. Кроме того, экспертным путем сформулированы дополнительные условия, влияющие на качество определения принадлежности писем к спаму.

Применение разработанной методики при классификации писем показало ее эффективность в сравнении с существующими методами фильтрации спама, а также работой подобных ЭС, что подтверждено процентными уровнями коэффициента ошибочного пропуска (3 %), коэффициента ошибочного отказа (1 %) и ошибки обобщения (2 %).

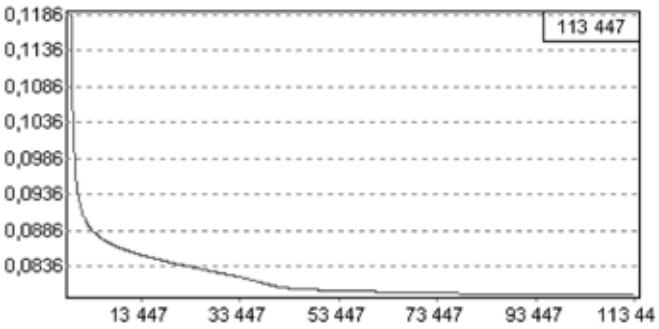

Рис. 4. Изменение ошибки при автоматической кластеризации значений входных параметров по предлагаемой методике

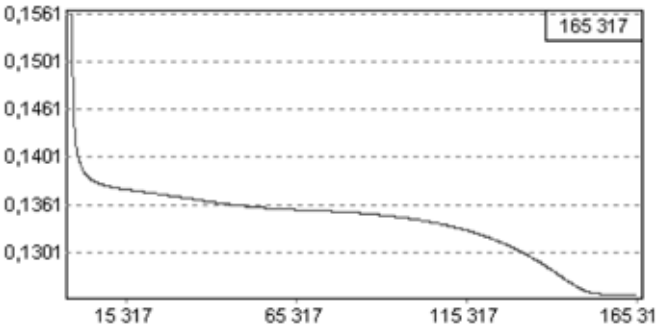

Рис. 5. Изменение ошибки при субъективном выборе числа градаций

Сравнение результатов обучения нейронной сети

Параметры	Определение числа градаций и форм ФП	
	Автоматически по предлагаемой методике	Субъективно на основе экспертной оценки
Ошибка выхода Время обучения Число эпох обучения	0,072 00:01:20 113 447	0,118 00:01:45 165 377

Кроме того, для апробации методики на базе программного комплекса "Нечеткая нейронная сеть" проанализировано влияние преднастройки и оптимизации на скорость и точность обучения ННС на примере данных медицинской диагностики [5]. Для этого использованы данные клинического, нейроортопедического, рентгенокомпьютерно-томографического обследования 230 женщин в возрасте от 15 до 92 лет и 180 мужчин в возрасте от 16 до 81 года с различными синдромами поясничного остеохондроза. Выборка данных включала совокупно более 50 000 значений количественных признаков течения остеохондроза по 79 параметрам.

Эффективность работы методики оценивалась по изменению ошибки выхода и времени обучения ННС при различных подходах к выбору числа градаций ЛП и форм их функций принадлежности.

На рис. 4 и 5 представлены типовые примеры зависимостей ошибки выхода ННС от числа циклов ее обучения при различных подходах к выбору числа градаций входных параметров.

При использовании методики точность обучения ННС превосходила точность обучения сети, в которой выбор числа градаций входных нейронов и форм их ФП определялся субъективно экспертом. Кроме того, в последнем случае нейронная сеть обучается нестабильно. Числовые характеристики процессов обучения приведены в таблице.

Рассмотренный пример иллюстрирует заметное улучшение качества обучения ННС при использовании предлагаемой методики и, как следствие, повышение точности аппроксимации экспериментальных данных.

Заключение

Путем преднастройки и оптимизации параметров ННС можно существенно повысить эффективность аппроксимации экспериментальных данных. Для решения данной задачи предложена методика, научная новизна которой заключается в применении специально разработанного алгоритма нечеткой кластеризации для выделения значений входных ЛП, исключающего недостатки существующих алгоритмов и имеющего практическую применимость. Методика позволяет до начала работы ННС автоматически выбрать объективно оптимальное число нечетких градаций значений входных нейронов и наилучшим образом описывающую их форму и начальные параметры функций принадлежности. Ее применение повышает эффективность нечеткой аппроксимации экспериментальных данных более чем на 30 %, что подтверждено результатами проведенной апробации.

Отметим, что предложенная методика может быть полезна для получения исходных данных при работе с нечеткими моделями, например при прогнозировании надежности алгоритмических процессов. Другое перспективное направление использования методики при построении ФП — формирование нечетких обучающих выборок для идентификации нелинейных зависимостей нечеткими базами знаний.

Список литературы

1. **Turing A. M.** Computing machinery and intelligence // Mind. 1950. № 59. С. 433—460.

2. **Куприянов М. С., Ярыгин О. Н.** Построение отношения и меры сходства нечетких объектов // Техническая кибернетика. 1988. № 3.

3. **Катасёв А. С.** Нейронечеткая модель и программный комплекс формирования баз знаний экспертных систем. Автореф. дисс. канд. техн. наук, Казань, 2006. 20 с.

4. **Светуньков С. Г.** Общая схема оценивания эконометрических моделей // Известия Санкт-Петербургского государственного университета экономики и финансов. 2002. № 3. С. 15—25.

5. **Подольская М. А., Нуриев З. Ш.** Компьютерно-томографическое исследование паравертебральных мышц на поясничном уровне при дистрофических вертеброгенных заболеваниях // Медицинская визуализация. 2004. № 4. С. 127—136.

УДК 004.94

П. Н. Девянин, д-р техн. наук,
доц., зам. зав. каф.,
Институт криптографии, связи и информатики,
e-mail: peter_devyanin@hotmail.com

Обзор семейства ДП-моделей безопасности логического управления доступом и информационными потоками в компьютерных системах

Описываются основные свойства семейства предлагаемых автором формальных моделей безопасности логического управления доступом и информационными потоками (ДП-моделей) в компьютерных системах с дискреционным, мандатным или ролевым управлением доступом. Анализируются направления развития и практического применения таких моделей.

Ключевые слова: *компьютерная безопасность, формальные модели, ДП-модели*

ДП-модели компьютерных систем с дискреционным или мандатным управлением доступом

Создание моделей безопасности логического управления доступом и информационными потоками (ДП-моделей) в компьютерных системах (КС) — одно из наиболее динамично развивающихся направлений современной теории компьютерной безопасности. Разработка и исследование свойств формальных моделей (формальная модель — модель, заданная на математическом или ином другом формализованном языке) может позволить реализовывать теоретически обоснованные подходы для обеспечения безопасности в существующих или перспективных КС, особенно в КС с высоким уровнем доверия к их безопасности [1].

В то же время сложившаяся практика проектирования защищенных КС нередко предполагает сначала разработку функциональной части системы защиты КС, а затем, как правило, неформальное или на основе неформальной модели (неформальная модель — упрощенное описание процесса или ситуации в терминах естественного языка) обоснование выполнения функциональной частью требований защиты информации в КС. При этом для администрирования и управления доступом защищенных КС также могут быть использованы подходы, не имеющие строгого теоретического обоснования. Однако, и это следует подчеркнуть, без формального моделирования логического управления доступом и информационными потоками разработка защищенных КС и обеспечение гарантий выполнения в них требований защиты информации во многих случаях не может быть эффективной.

Основными известными автору формальными моделями, которые потенциально могут быть применены для исследования безопасности КС с дискреционным, мандатным или ролевым управлением доступом, являются следующие модели (данные модели детально рассмотрены в работе [3]):

- модель Харрисона—Руззо—Ульмана и ее развитие — модель типизированной матрицы доступов (ТМД) [16, 22];
- модель *Take-Grant* и ее основные расширения [13, 15];
- классическая модель Белла—ЛаПадулы и ее интерпретации (в том числе, интерпретация "безопасность переходов"), модель мандатной политики целостности информации Биба [11, 12, 18];
- модель систем военных сообщений (СВС) [17];
- автоматная, программная и вероятностная модели безопасности информационных потоков [14, 19];
- базовая модель ролевого управления доступом (RBAC) и ее расширения, а именно — модель ролевого администрирования и модель мандатного ролевого управления доступом [20, 21];
- субъектно-ориентированная модель изолированной программной среды (ИПС) [10].

В данных формальных моделях использованы оригинальные определения основных элементов и механизмов КС и, как правило, не учтены следующие существенные особенности функционирования современных КС:

- возможность кооперации части субъектов при передаче прав доступа и создании информационных потоков;
- возможность реализации в КС доверенных и недоверенных субъектов с различными условиями функционирования;
- возможность противодействия доверенными субъектами КС передаче прав доступа или соз-

данию информационных потоков недоверенными субъектами;
- различие условий реализации в КС информационных потоков по памяти и по времени;
- наличие в КС иерархической структуры сущностей и возможность ее использования при создании информационных потоков по времени;
- возможность изменения функциональности субъекта при реализации информационного потока по памяти на функционально ассоциированные с ним сущности или от параметрически ассоциированных с ним сущностей;
- необходимость в ряде случаев определения различных правил управления доступом и информационными потоками для распределенных компонентов КС.

В целях обеспечения возможности теоретического анализа условий утечки прав доступа и реализации запрещенных информационных потоков по памяти или по времени с учетом приведенных существенных особенностей современных КС автором построено семейство формальных моделей безопасности логического управления доступом и информационными потоками (сокращенно, семейство ДП-моделей) [4]. Первоначально в состав этого семейства вошло 10 ДП-моделей КС с дискреционным или мандатным управлением доступом.

Основой всех моделей семейства является базовая ДП-модель, построенная с применением положений расширенной модели *Take-Grant*, модели Белла—ЛаПадулы, модели СВС и субъектно-ориентированной модели ИПС. При этом использован классический подход, состоящий в том, что каждая моделируемая КС представляется абстрактной системой, каждое состояние которой описывается графом доступов, а любой переход системы из состояния в состояние осуществляется с помощью одного из правил преобразования графов доступов.

В рамках базовой ДП-модели рассматриваются множества субъектов, сущностей (объектов или контейнеров), видов прав доступа, доступов и информационных потоков, задаются иерархия сущностей и 16 монотонных и немонотонных правил преобразования состояний. При этом обосновываются необходимые и достаточные условия передачи в КС прав доступа или реализации информационных потоков по памяти или по времени.

Для анализа КС, в которых все субъекты являются либо доверенными, либо недоверенными, когда доверенные субъекты не кооперируют с недоверенными при передаче прав доступа или реализации информационных потоков, построена ДП-модель без кооперации доверенных и недоверенных субъектов (БК ДП-модель). На ее основе строится ДП-модель с блокирующими доступами доверенных субъектов (БД ДП-модель), в рамках которой анализируются условия реализации в КС запрещенных информационных потоков по времени для случая, когда доверенные субъекты препятствуют использованию недоверенными субъектами иерархии сущностей для создания таких информационных потоков.

На основе БК ДП-модели строится ДП-модель с функционально ассоциированными с субъектами сущностями (ФАС ДП-модель), которая позволяет анализировать условия получения недоверенным субъектом права доступа владения к доверенному субъекту с использованием реализации недоверенным субъектом информационного потока по памяти к сущности, функционально ассоциированной с доверенным субъектом.

На основе ФАС ДП-модели построена ДП-модель для политики безопасного администрирования (ПБА ДП-модель). Такая модель позволяет анализировать условия обеспечения защиты распределенных КС:
- от захвата нарушителем (недоверенным субъектом) контроля над компьютером, на котором нарушитель разместил свои ресурсы (запустил процесс или разместил файлы);
- от захвата нарушителем прав доступа и привилегий доверенного субъекта, обратившегося (с целью получить данные файлов локально или по сетевым коммуникационным каналам) к компьютеру, на котором разместил свои ресурсы нарушитель.

Кроме ПБА ДП-модели на основе ФАС ДП-модели построена ДП-модель для политики абсолютного разделения административных и пользовательских полномочий (ПАР ДП-модель). Эта модель направлена на исследование условий обеспечения безопасности рабочих станций пользователей распределенных КС для случая, когда при наличии на рабочей станции активных недоверенных субъектов возможно блокирование использования любыми субъектами административных прав доступа или привилегий.

Для анализа безопасности КС с мандатным управлением доступом построена мандатная ДП-модель. В рамках данной модели показано, что обеспечения безопасности состояний и функции переходов в смысле Белла—ЛаПадулы (в смысле определений *ss*- и *-свойств безопасности) недостаточно для противодействия возможности реализации запрещенных информационных потоков двух видов:
- информационных потоков по времени от сущностей с высоким уровнем конфиденциальности информации к сущностям с низким уровнем конфиденциальности информации;
- информационных потоков по памяти от недоверенных субъектов с низким уровнем доступа к сущностям, функционально ассоциированным с субъектами с высоким уровнем доступа.

Для анализа условий, выполнение которых в КС с мандатным управлением доступом позволит предотвратить возможность реализации этих запрещенных информационных потоков, на основе мандатной, БД и ФАС ДП-моделей построены следующие три ДП-модели:

- мандатная ДП-модель с блокирующими доступами доверенных субъектов (БДМ ДП-модель), в которой анализируются условия реализации в КС запрещенных информационных потоков для случая, когда доверенные субъекты препятствуют использованию недоверенными субъектами иерархии сущностей для создания информационных потоков по времени;
- мандатная ДП-модель с отождествлением порожденных субъектов (ОСМ ДП-модель), в которой при определении *-свойства безопасности рассматриваются доступы недоверенного субъекта вместе со всеми доступами порожденных им субъектов;
- мандатная ДП-модель компьютерных систем, реализующих политику строгого мандатного управления доступом (ПСМ ДП-модель), в рамках которой недоверенному субъекту разрешается получать любые виды доступа только к сущностям с уровнем конфиденциальности, совпадающим с его уровнем доступа.

С использованием БД, ФАС, ПБА, ПАР, мандатной, БДМ, ОСМ и ПСМ ДП-моделей в работе [4] описаны и теоретически обоснованы пять методов, реализация которых в КС с дискреционным или мандатным управлением доступом позволяет предотвратить возможность возникновения некоторых видов запрещенных информационных потоков по памяти или по времени.

ДП-модели компьютерных систем с ролевым управлением доступом

В известной автору литературе основное внимание при исследовании ролевого управления доступом, как правило, уделяется вопросам его администрирования или разработке моделей, адаптированных к условиям функционирования конкретных КС. Анализу безопасности информационных потоков, возникающих в результате реализации субъектами (субъект-сессиями) доступов к сущностям, уделяется недостаточно внимания. Кроме того, основные модели ролевого управления доступом не содержат описания правил перехода КС из состояния в состояние. В то же время результаты проведенных на практике экспериментов показали, что нарушитель, используя часто разрешенные политикой безопасности КС доступы к сущностям, может эффективно реализовать запрещенные информационные потоки по памяти или по времени. Отсутствие при этом четких правил перехода КС из состояния в состояние может привести к разработке и использованию для анализа безопасности КС неадекватных ей формальных моделей.

Для исследования безопасности КС с учетом особенностей ролевого управления доступом на основе семейства ролевых моделей *RBAC* и семейства ДП-моделей КС с дискреционным или мандатным управлением доступом автором построена базовая ролевая ДП-модель (БР ДП-модель) [5, 6]. Данная модель ориентирована на анализ в КС с ролевым управлением доступом условий передачи прав доступа ролей и реализации информационных потоков по памяти и по времени.

В КС с ролевым управлением доступом право доступа к сущности может быть получено субъект-сессией только через обладание ролью, содержащей данное право. Реализация субъект-сессией информационного потока по памяти на сущность, функционально ассоциированную с другой субъект-сессией, позволит первой субъект-сессии получить контроль над второй субъект-сессией, включая возможность использовать права доступа ее ролей. При этом множество текущих ролей первой субъект-сессии, как правило, останется неизменным (кроме того, изменение множества текущих ролей может быть заблокировано реализованным в КС механизмом статических и, особенно динамических, ограничений). Таким образом, в рамках БР ДП-модели, кроме ролей, прав доступа ролей и возможностей осуществить действия, которыми явно обладают субъект-сессии, рассматриваются фактические роли, фактические права доступа ролей и фактические возможности осуществления действий, которыми обладают субъект-сессии за счет получения контроля над другими субъект-сессиями. При этом фактические роли, права доступа ролей и возможности учитываются в БР ДП-модели при описании условий и результатов применения правил преобразования состояний (рис. 1).

В настоящее время в рамках БР ДП-модели не удалось завершить исследование КС с ролевым управлением доступом, на условия функционирования которых не наложено ограничений. В связи с этим обстоятельством анализ необходимых и достаточных условий передачи прав доступа выполнен для двух случаев. Первый случай, когда в КС существуют только две субъект-сессии двух пользователей. Второй случай, когда в КС функционирует произвольное число субъект-сессий и они не получают доступа владения друг к другу с использованием информационных потоков по памяти к функционально ассоциированным с субъект-сессиями сущностям.

В дальнейшем автором планируются разработка и исследование ролевых ДП-моделей по следующим направлениям:

Рис. 1. Зависимость условий и результатов применения правил преобразования состояний БР ДП-модели

- разработка моделей, адекватных условиям функционирования существующих КС;
- построение ДП-моделей существующих КС для формального анализа их безопасности;
- анализ условий кооперации доверенных и недоверенных субъект-сессий при реализации информационных потоков;
- построение алгоритмов замыкания графа доступов, описывающего состояние системы;
- анализ условий передачи прав доступа ролей для случая произвольного числа субъект-сессий, а также анализ условий возникновения информационных потоков по памяти или по времени;
- исследование способов использования динамических ограничений;
- исследование возможности реализации мандатного ролевого управления доступом с применением динамических ограничений, не позволяющих порождать запрещенные информационные потоки по времени.

Развитие семейства ДП-моделей

Развитие семейства ДП-моделей в настоящее время осуществляется по двум направлениям:

- построение ДП-моделей типовых практически значимых или перспективных КС;
- построение ДП-моделей, содержащих элементы принципиально новые по сравнению с элементами уже разработанных ДП-моделей.

По первому направлению построены ДП-модели для нескольких реальных КС, в том числе ДП-модель веб-системы [8] и ДП-модель веб-системы на основе СУБД [9]. В рамках двух отмеченных моделей осуществлен анализ защищенности типовых веб-систем от угроз реализации атак вида межсайтового скриптинга (*Cross Site Scripting, XSS*) и вида *SQL*-инъекции.

По второму направлению в работе [7] рассмотрен случай, когда в КС могут существовать параметрически ассоциированные с субъектами сущности, реализация от которых информационных потоков по памяти к недоверенным субъектам позволяет им получить контроль над другими субъектами системы, в том числе доверенными. Например, возможно получение субъектом-нарушителем права доступа на чтение к конфигурационному файлу операционной системы, в котором хранится пароль или хеш-образ пароля доверенного субъекта. Получение такого права позволяет субъекту-нарушителю получить контроль над доверенным субъектом. В данном случае конфигурационный файл будет являться сущностью, параметрически ассоциированной с доверенным субъектом. Таким образом, в работе [7] на основе ФАС ДП-модели построена ДП-модель с функционально или параметрически ассоциированными с субъектами сущностями (ФПАС ДП-модель). В рамках такой модели выполнен теоретический анализ условий утечки прав доступа и реализации запрещенных информационных потоков по памяти.

Также по второму направлению на основе ФАС и ФПАС ДП-моделей в работе [2] построена ДП-модель файловых систем (ФС ДП-модель). В данной модели анализируется характерный для файловых систем новый вид доверенных субъектов — потенциальных доверенных субъектов. Из потенциальных доверенных субъектов могут быть созданы доверенные субъекты, реализующие доступ к сущностям, защищенным механизмами файловых систем (например, механизмами файловой системы *EFS* в среде ОС семейства *Windows XP*/2003/*Vista*). Создать доверенного субъекта из потенциального доверенного субъекта могут доверенные или недоверенные субъекты, имеющие доступ на чтение к сущностям, параметрически

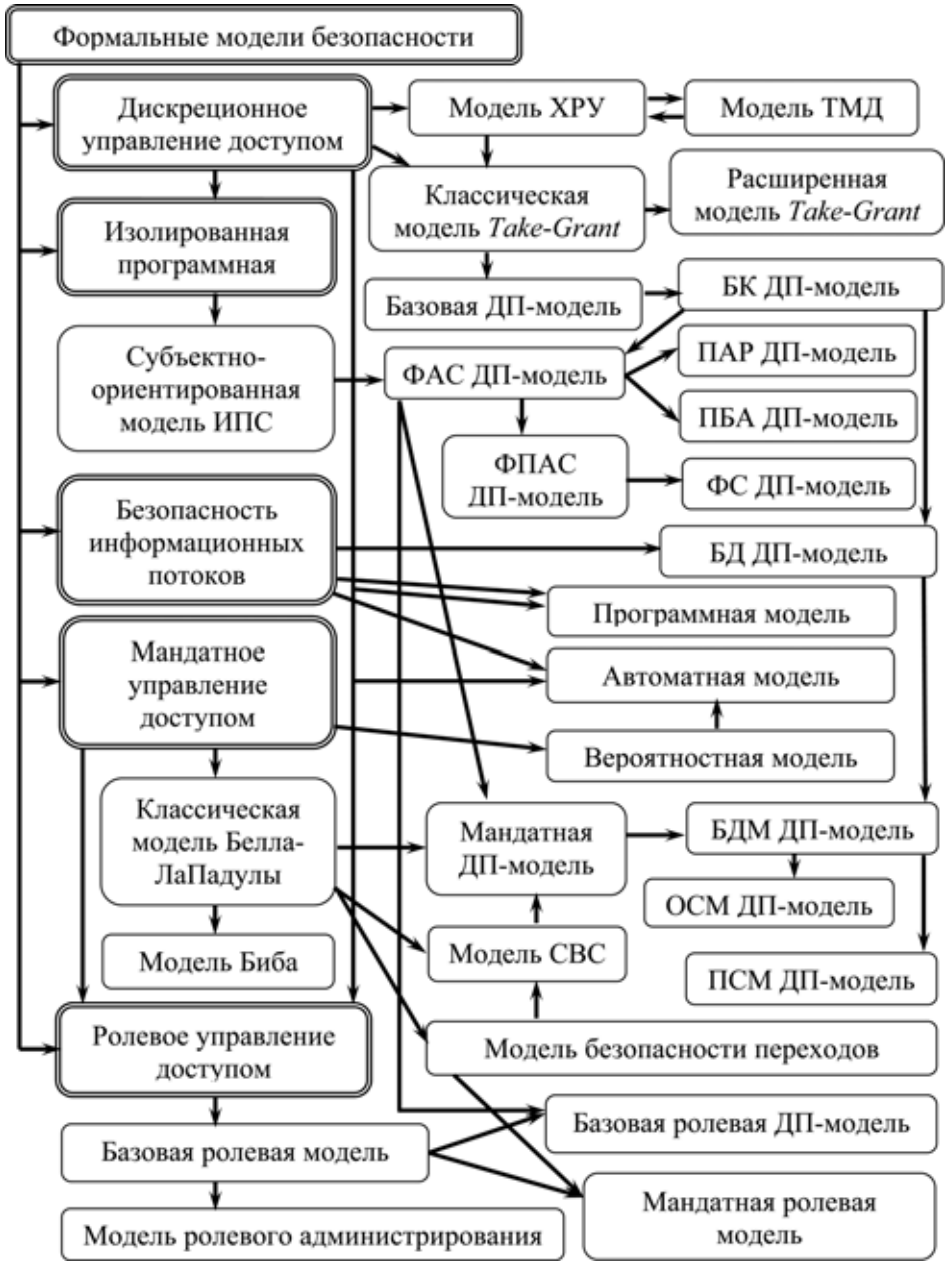

Рис. 2. Классификация и взаимосвязь положений основных классических формальных моделей безопасности КС и семейства ДП-моделей

ассоциированным с потенциальными доверенными субъектами. В рамках ФС ДП-модели проведен анализ условий реализации информационных потоков. В том числе, приведены и обоснованы достаточные условия, при выполнении которых в КС невозможна реализация запрещенных информационных потоков по памяти от сущностей, защищенных механизмами файловых систем.

Выводы

В целях анализа условий передачи прав доступа или реализации информационных потоков по памяти или по времени на основе классических формальных моделей безопасности КС построено семейство ДП-моделей КС с дискреционным, мандатным или ролевым управлением доступом. При этом в ДП-моделях учтены несколько существенных особенностей функционирования современных КС. Классификация и взаимосвязь положений основных классических формальных моделей безопасности КС и семейства ДП-моделей показаны на рис. 2.

В настоящее время семейство ДП-моделей расширяется за счет включения в него ДП-моделей типовых реальных КС. Разрабатываются ДП-модели, включающие принципиально новые элементы. Кроме того, с использованием ДП-моделей проводятся исследования наиболее перспективного вида логического управления доступом — ролевого управления доступом.

Опыт построения и развития семейства ДП-моделей показал, что они могут быть эффективно использованы для анализа безопасности существующих или перспективных КС.

Список литературы

1. **Безопасность** информационных технологий. Критерии оценки безопасности информационных технологий // Руководящий документ (ГОСТ Р ИСО/МЭК 15408). М.: Гостехкомиссия России, 2002.

2. **Буренин П. В.** Подходы к построению ДП-модели файловых систем // Прикладная дискретная математика. 2009. № 1 (3). С. 93—112.

3. **Девянин П. Н**. Модели безопасности компьютерных систем: учеб. пособие для студ. высш. учеб. заведений. М.: Академия, 2005. 144 с.

4. **Девянин П. Н**. Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006. 176 с.

5. **Девянин П. Н.** Базовая ролевая ДП-модель // Прикладная дискретная математика. 2008. № 1 (1). С. 64—70.

6. **Девянин П. Н.** О разработке моделей безопасности информационных потоков в компьютерных системах с ролевым управлением доступом // Материалы Третьей международной научной конференции по проблемам безопасности и противо-

действия терроризму. МГУ им. Ломоносова. 25—27 октября 2007 г. М.: МЦНМО, 2008. С. 261—265.

7. **Колегов Д. Н.** ДП-модель компьютерной системы с функционально и параметрически ассоциированными с субъектами сущностями // Вестник Сибирского государственного аэрокосмического университета имени академика М. Ф. Решетнева. 2009. Вып. 1 (22). Часть 1. С. 49—54.

8. **Назаров И. О**. Анализ безопасности веб-систем в условиях реализации уязвимости класса межсайтового скриптинга // Проблемы информационной безопасности. Компьютерные системы. 2007. Вып. 2. С. 105—117.

9. **Назаров И. О.** Обеспечение безопасности управления доступом и информационными потоками в веб-системе на основе СУБД // Вестник Казанского государственного технического университета им. А. Н. Туполева. 2008. Вып. 2. С. 56—59.

10. **Щербаков А. Ю.** Современная компьютерная безопасность. Теоретические основы. Практические аспекты: учеб. пособие. — М.: Книжный мир, 2009. 352 с.

11. **Bell D. E., LaPadula L. J.** Secure Computer Systems: Unified Exposition and Multics Interpretation. Bedford: MITRE Corp., 1976.

12. **Biba K.** Integrity Considerations for Secure Computer Systems // Technical Report MTR-3153. Bedford: MITRE Corp. 1977.

13. **Bishop M.** Computer Security: art and science. 2002. 1084 p.

14. **Castano S., Fugini M. G., Martella G., Samarati P.** Database Security. Addison Wesley Publishing Company, ACM Press, 1995. 456 p.

15. **Frank J., Bishop M.** Extending the Take-Grant Protection System. University of California at Davis, 1984.

16. **Harrison M., Ruzzo W., Ullman J.** Protection in operating systems // Communication of ACM. 1976. 19 (8). P. 461—471.

17. **Lanawehrm E., Heitmeyer L., McLean J**. A Security Model for Military Message Systems // ACM Trans. On Computer Systems. 1984. Vol. 9, № 3.

18. **McLean J.** The Specification and Modeling of Computer Security. // Computer. 1990. Vol. 23. N 1.

19. **McLean J., John D.** Security Models and Information Flow // Proc. of 1990 IEEE Symposium on Research in Security and Privacy. IEEE Press, 1990.

20. **Sandhu R.** Rationale for the RBAC96 family of access control models // Proc. of the 1st ACM Workshop on Role-Based Access Control. — ACM, 1997.

21. **Sandhu R.** Role-Based Access Control, Advanced in Computers. Vol. 4. // Academic Press, 1998.

22. **Sandhu R.** The typed access matrix model // Proc. of the IEEE Symposium on Research in Security and Privace, Oakland, CA. May 1992. P. 122—136.

УДК 004.056.53

А. П. Типикин, д-р техн. наук,
А. С. Глазков, аспирант,
e-mail: Room65@yandex.ru,
КурскГТУ

Метод и функциональная организация контроля обращений и закрытия доступа к секторам файлов при хищении накопителя информации

Описаны метод и технологическая схема извлечения метаданных устройства хранения данных, позволяющие существенно снизить вероятность восстановления пользовательских данных злоумышленником при хищении накопителя информации. Приведена оценка объема основных метаданных, извлекаемых из устройства хранения данных.

Ключевые слова: *жесткий магнитный диск, носитель информации, хищение, информация, устройство, ограничение, доступ, хранение, данные, метаданные, сектор, раздел, файл, главная таблица файлов, основная загрузочная запись*

Для ограничения доступа к данным, хранимым в ЭВМ, используются программные или программно-аппаратные средства. Современные программно-аппаратные системы ограничения несанкционированного доступа (СОНД) к данным, записанным на жестком магнитном диске ЭВМ, состоят из аппаратной части, непосредственно реализующей функции ограничения доступа, и управляющего программного обеспечения (УПО), осуществляющего взаимосвязь пользователя с устройством [1].

Примером СОНД может служить встроенное в контроллер накопителя устройство ограничения доступа (УОД) к секторам жесткого магнитного диска (ЖМД), поддерживаемое УПО [2]. В этом устройстве атрибуты защиты секторов файлов переносятся из программного уровня ЭВМ на аппаратный уровень контроллера накопителя информации, что существенно снижает вероятность несанкционированного доступа к ним и в конечном счете способствует обеспечению приватности файлов и повышению надежности всей компьютерной системы в целом без больших затрат ресурсов ЭВМ. СОНД [2] гарантирует высокую стойкость защиты данных при физической недоступности накопителя на жестких магнитных дисках (НЖМД), находящегося внутри рабочей станции.

Однако вероятность удачного выполнения атаки может существенно повыситься, если НЖМД окажется в руках злоумышленника и будет подключен к его рабочей станции. Угроза "хищение НЖМД" может наблюдаться в следующих случаях:

- размещении серверов в стороннем дата-центре (*collocation*);
- отправки серверов или жестких дисков в ремонт;
- перевозки компьютеров из одного офиса в другой, например при переезде;

- утилизации компьютеров, серверов, жестких дисков и лент;
- хранении НЖМД в специальном депозитарии (*off-site storage*);
- перевозки НЖМД, например в депозитарий;
- кражи или потери НЖМД.

Если СОНД содержит УОД, встроенное в контроллер [2], воспользоваться стандартным интерфейсом НЖМД для атаки на данные невозможно, так как устройство [2] может функционировать без операционной системы и управляющего программного обеспечения. Это существенно уменьшает возможные случаи реализации угрозы "хищение НЖМД", так как для обхода защиты УОД [2] у злоумышленника остаются следующие два варианта атаки:

- замена модифицированного контроллера на стандартный с другого аналогичного НЖМД;
- извлечение магнитных дисков из НЖМД и попытка их чтения на специализированном оборудовании [2, 4].

Однако и в названных случаях СОНД, содержащая УОД, встроенное в НЖМД [2], может быть использована для защиты данных на основе следующего метода закрытия доступа к секторам файлов при хищении накопителя информации.

Под методом закрытия доступа к секторам файлов мы будем понимать метод защиты от копирования злоумышленником данных легитимного пользователя. При выполнении названного копирования прежде, чем получен доступ к данным, необходимо проанализировать носитель информации. Процедуры такого анализа можно представить схемой, приведенной на рис. 1. С ЖМД считывается поток информации, который анализируется, и в результате получают информацию о томах/разделах. Каждый том/раздел анализируется в отдельности: определяется его местоположение, размер на ЖМД, тип файловой системы. После анализа файловой системы тома получают информацию о хранимых в нем на ЖМД файлах: имена, размеры, местоположение. Далее полученные файлы анализируются на прикладном уровне.

По своей смысловой нагрузке данные, хранящиеся в НЖМД, могут быть разделены на два класса: метаданные и данные пользователя. Практически любой анализ физического носителя, томов, разделов, файловой системы начинается на уровне метаданных.

Метаданными принято называть "данные о данных". Однако их значение распространяется, помимо описания состава данных, их структуры (формата) представления, места хранения и других признаков описания, также на поддерживающие их информационные системы, технологии, методы доступа, пользователей и т. д. [5]. Метаданные содержат всю необходимую информацию

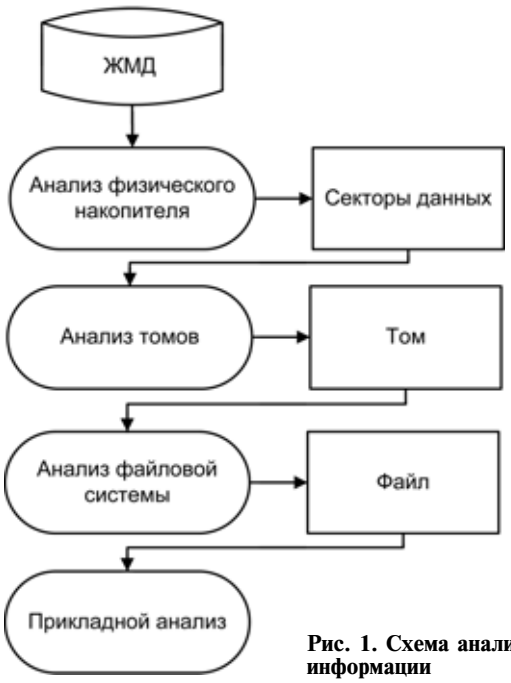

Рис. 1. Схема анализа накопителя информации

о разделах, файловых системах и пользовательских данных жесткого диска. Во время включения компьютера BIOS (базовая система ввода-вывода) материнская плата считывает метаданные и по ним определяет, сколько разделов содержит жесткий диск и с какого из них необходимо загружать операционную систему (ОС). В свою очередь ОС при работе с файлами обращается к метаданным файловой системы (ФС), откуда она считывает всю необходимую информацию для выполнения заданной операции над файлом. Таким образом, метаданные играют решающую роль в работе компонентов ОС, отвечающих за функционирование ФС и вычислительной системы в целом.

Программы восстановления/поиска данных, которыми могут воспользоваться злоумышленники, также применяют алгоритмы анализа метаданных и их сигнатур [6]. Если после окончания работы на ЭВМ легитимный пользователь с помощью средств СОНД из носителя извлечет наиболее важные метаданные, то злоумышленник не сможет воспользоваться записанными на ЖМД данными применением стандартных средств ОС, если даже подключит похищенное НЖМД к своей рабочей станции. Для считывания информации ему придется применять специализированные программы восстановления данных (*data recovery*), алгоритмы большинства из которых основаны на анализе метаданных, что делает их в данном случае малоэффективными. Считывание данных напрямую из секторов ЖМД без использования метаданных ФС — трудоемкий и долговременный процесс, который не дает стопроцентной гарантии получения нужного результата и усложняется из-за фрагментации диска.

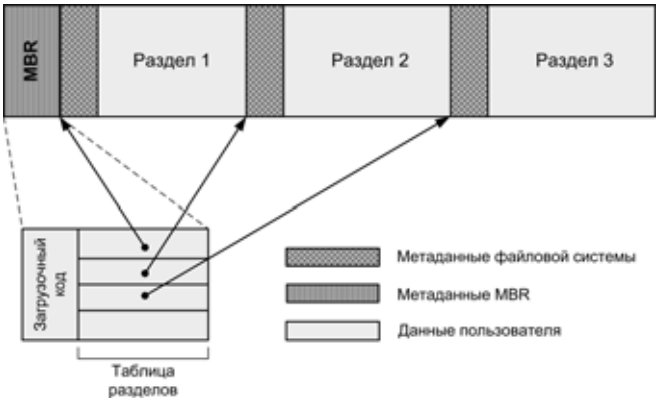

Рис. 2. Логическая структура ЖМД

Таблица 1

Внутренняя структура MBR

Смещение	Размер поля, байт	Описание
0 × 000	446	Загрузочная запись (MSB)
0 × 1BE	16	Описатель раздела 1
0 × 1CE	16	Описатель раздела 2
0 × 1DE	16	Описатель раздела 3
0 × 1EE	16	Описатель раздела 4
0 × 1FE	2	Сигнатура таблицы разделов (значение 0×AA55)

Рассмотрим более подробно метаданные ЖМД (рис. 2).

К метаданным ЖМД относятся:

- основная загрузочная запись MBR (*Master Boot Record*);
- метаданные файловой системы (ФС), содержащие информацию обо всех файлах и каталогах.

Основная загрузочная запись (MBR) находится в первом 512-байтовом секторе диска (с абсолютным номером 0). MBR содержит загрузочный код BIOS (*ROM Bootstrap Routine*), который при загрузке с жесткого диска считывается и инициализирует загрузку в память ЭВМ первого физического сектора на активном разделе диска, называемого загрузочным сектором (*Boot Sector*), а также таблицу разделов (*Partition Table*) и сигнатуру (0xAA55) (табл. 1).

Таблица разделов содержит до трех первичных разделов и один первичный расширенный раздел. Первичным разделом файловой системы называется раздел, представленный записью в MBR и содержащий файловую систему или другие структурированные данные. Первичным расширенным разделом называется раздел, представленный записью в MBR и содержащий вторичные разделы. Каждый вторичный раздел содержит свою собственную вторичную MBR (SMBR), имеющую структуру, аналогичную MBR, но загрузочная запись в нем отсутствует (заполнена нулями), а из четырех полей описателей используется только два. Для таблицы разделов и загрузочного кода обычно хватает одного сектора, но, как правило, для MBR и расширенных разделов выделяются 63 сектора, потому что разделы должны начинаться на границе цилиндра [6].

Таким образом, если легитимный пользователь извлечет из ЖМД на свой ключевой внешний носитель информации все содержимое первых секторов с 0-го по 63-й, злоумышленник не сможет в короткий срок установить число и местоположение разделов ЖМД.

Извлечение из ЖМД только данных MBR о списке расширенных разделов недостаточно, так как служебные поля ФС имеют специфические идентификаторы и сигнатуры, по которым сканирующая программа сможет определить положение области метаданных ФС и в результате — положение самих данных.

Рассмотрим метаданные файловой системы на примере наиболее распространенной системы NTFS (*New Technology File System*) [5, 6]. Самый главный файл NTFS — MFT (*Master File Table*) и общая таблица файлов. Каждый файл и каталог представлен в MFT как минимум одной записью размером 1 Кбайт. Раздел NTFS условно делится на две части (рис. 3). Первые 12 % диска отводятся под MFT-зону — пространство, в которое растет метафайл MFT. Запись каких-либо данных в эту область невозможна. MFT-зона всегда держится пустой. Это делается для того, чтобы MFT не фрагментировался при своем росте. Остальные 88 % диска представляют собой обычное пространство для хранения файлов.

В документации Microsoft говорится, что резервируются только первые 16 служебных записей MFT, но на практике выделение записей пользовательским файлам и каталогам начинается с записи 24. Записи 17—23 иногда используются в тех ситуациях, когда зарезервированных записей не хватает. Следует отметить, что вторая копия первых трех записей для повышения надежности хранится ровно посередине диска.

Лучшим способом защиты от угрозы "хищение НЖМД" было бы извлечение из ЖМД всего MFT-метафайла, так как он содержит всю информацию о файлах пользователя. К сожалению, он может занимать до 12 % пространства раздела. Учитывая, что емкость современного ЖМД достигает 1,5 Тбайт, а раздел NTFS может занимать

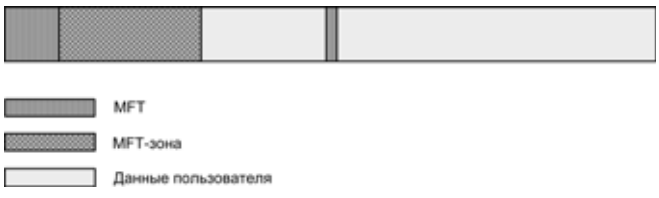

Рис. 3. Структура NTFS-раздела

все пространство диска, размер MFT-метафайла может достигать 180 Гбайт. Извлечение такого объема служебной информации с ЖМД потребует больших затрат времени и приведет к увеличению стоимости внешнего портативного накопителя информации, являющегося ключевой картой пользователя. Целесообразно извлекать только служебные записи MFT: (0—23) записи и три записи посередине раздела, так как они содержат специфические сигнатуры, используемые сканирующими программами злоумышленников для определения местоположения метаданных ФС [6].

Аналогично можно определить места хранения на ЖМД метаданных любого из типов ФС. Разные диапазоны их адресов на ЖМД могут быть сохранены на ключевом внешнем носителе информации, например на флэш-карте, и использованы в процессе их извлечения после определения по MBR типа ФС защищаемого раздела ЖМД.

Оценим объем основных метаданных, извлекаемых из НЖМД:

- основной загрузочной записи MBR;
- списка записей вторичных разделов SMBR;
- списка служебных записей метаданных ФС.

Расчет объема извлекаемых метаданных V'_{MD} может быть выполнен по формуле

$$V'_{MD} = V_{MBR} + n_{SMBR}V_{SMBR} + n_{FS}V_{FS}, \qquad (1)$$

где V_{MBR} — объем, занимаемый *MBR*; n_{SMBR} — число вторичных разделов; V_{SMBR} — объем, занимаемый *SMBR*; n_{FS} — число разделов с ФС; V_{FS} — объем, занимаемый служебными записями метаданных ФС.

Структура MBR аналогична SMBR и занимает 512 байт. Размер служебных записей метаданных V_{FS} зависит от типа файловой системы. Для ФС типа NTFS необходимо извлекать 0—23 записи и три записи посередине раздела, каждая из которых занимает 1 Кбайт.

Для восстановления метаданных на накопителе НЖМД после включения ЭВМ необходимо на ключевом внешнем накопителе пользователя, кроме извлеченных метаданных, сохранять адреса их секторов на ЖМД. Так как извлекаемые метаданные не фрагментируются, то достаточно сохранять только адреса начальных секторов, причем для основной загрузочной записи MBR хранить такой адрес не обязательно, поскольку она всегда хранится в самом первом секторе диска с абсолютным номером 0. Для хранения адреса сектора ЖМД достаточно 4 байт. Таким образом, формула (1) после введения в нее информации об адресах начальных секторов примет следующий вид:

$$V''_{MD} = V_{MBR} + n_{SMBR}V_{SMBR} + n_{FS}V_{FS} + (n_{SMBR} + n_{FS})V_{SS},$$

где V_{SS} — размер адреса сектора.

Для того чтобы специальное управляющее программное обеспечение смогло определить именно то устройство хранения данных (НЖМД), которому принадлежат извлекаемые метаданные, необходимо еще сохранять идентификатор НЖМД. В качестве идентификатора можно использовать серийный номер производителя, хранящийся в паспорте ЖМД [8]. Итоговая формула для подсчета объема извлекаемых метаданных будет выглядеть следующим образом:

$$V_{MD} = V_{MBR} + n_{SMBR}V_{SMBR} + n_{FS}V_{FS} + (n_{SMBR} + n_{FS})V_{SS} + V_{IDHDD}, \qquad (2)$$

где V_{IDHDD} — размер идентификатора НЖМД.

В табл. 2 приведены результаты расчетов по формуле (2) для трех логических конфигураций НЖМД, имеющих разное число n_{SMBR} и n_{FS}.

Из табл. 2 видно, что даже если НЖМД имеет три расширенных раздела и пять разделов NTFS, объем извлекаемых метаданных составляет около 140 Кбайт, что является несущественным по сравнению с емкостью ключевых современных внешних накопителей данных, например флэш-памятей.

Рассмотрим технологическую схему и алгоритмы работы УПО для закрытия доступа к секторам с помощью СОНД [2] при хищении устройства хранения данных.

Процедуры извлечения и восстановления метаданных, осуществляемые специальным УПО, должны выполняться с помощью альтернативной ОС (АОС) до загрузки основной операционной системы. Современные ЭВМ позволяют выполнять предварительную загрузку АОС с внешних носителей информации, в том числе и с флэш-памяти.

Существует много вариантов таких упрощенных ОС, которые занимают около 20—25 Мбайт

Рис. 4. Архитектура системы извлечения метаданных

Таблица 2

Результаты расчетов метаданных НЖМД, байт

n_{SMBR}	n_{FS}	Объем извлекаемых метаданных НЖМД, байт
1	3	84 004
2	4	112 172
3	5	140 340

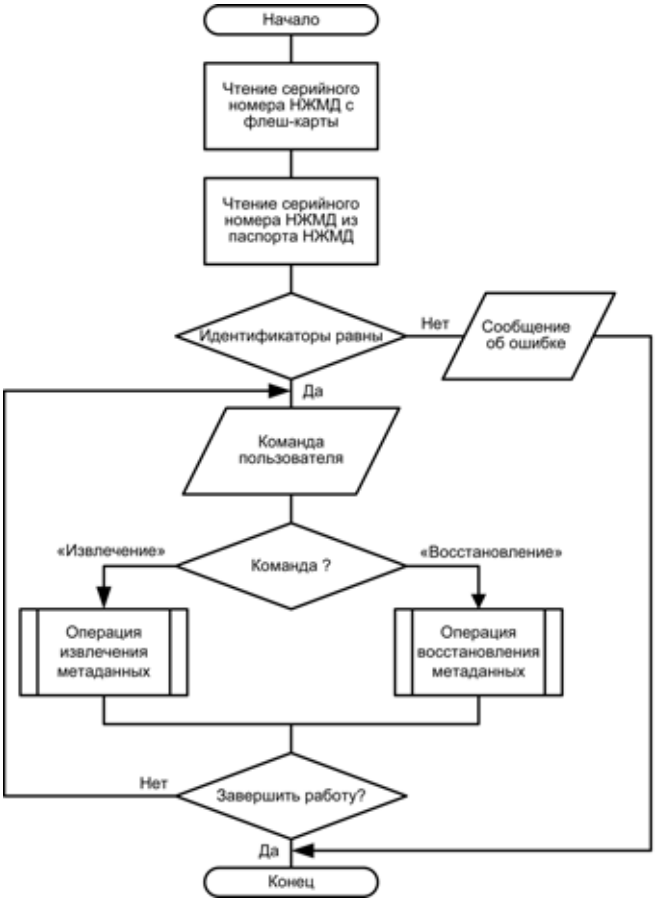

Рис. 5. Алгоритм работы УПО по извлечению/восстановлению метаданных НЖМД (начало)

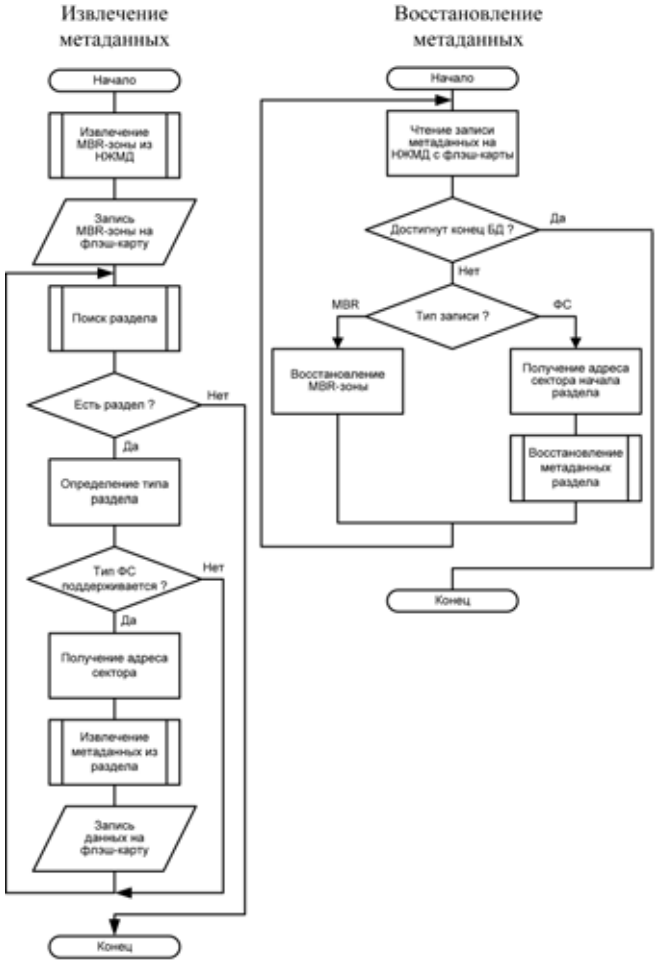

Рис. 6. Алгоритм работы УПО по извлечению/восстановлению метаданных НЖМД (окончание)

на носителе информации [9]. Это позволяет использовать в качестве индивидуального переносного пользовательского ключевого накопителя стандартную флэш-память с USB-интерфейсом.

На рис. 4 представлена архитектура системы извлечения метаданных.

На рис. 5 представлен алгоритм работы УПО по извлечению/восстановлению метаданных НЖМД.

Алгоритм состоит из трех частей:

- главная подпрограмма УПО по извлечению/восстановлению метаданных НЖМД (рис. 5), выполняющая общие для всех операций функции;
- подпрограмма извлечения метаданных НЖМД;
- подпрограмма восстановления метаданных НЖМД (рис. 6).

На рис. 6 условное обозначение БД — это база данных, содержащая метаданные, извлеченные из ЖМД.

Разработанный метод извлечения метаданных НЖМД позволяет существенно снизить вероятность восстановления пользовательских данных злоумышленником при хищении накопителя информации, что повышает приватность хранения файлов. Данный метод сравнительно прост и не требует применение специальных дополнительных аппаратных средств, что позволяет использовать его как в системах ограничения несанкционированного доступа, так и совместно со стандартной системой шифрования данных ОС. Если применить его в программно-аппаратной СОНД, основанной на УОД [2], восстановление метаданных на ЖМД станет нецелесообразным, так как в этом случае достаточно переписывать их при включении ЭВМ только в оперативную память УОД. Это позволяет снизить затраты времени на восстановление и последующие извлечения метаданных, а также предотвратить возможность обращения злоумышленника к ЖМД, похищенному в момент отлучки легитимного пользователя с рабочего места, так как при этом на ЖМД всегда будут отсутствовать метаданные.

Список литературы

1. **Романец Ю. В., Тимофеев П. А., Шаныгин В. Ф.** Защита информации в компьютерных системах / Под ред. В. Ф. Шаныгина. 2-е изд., перераб. и доп. М.: Радио и связь, 2001. 376 с.
2. **Патент 2359317 РФ, МПК7 G 06 F 12/14.** Устройство ограничения доступа к секторам жесткого диска / А. С. Глазков, А. П. Типикин, М. О. Таныгин; заявитель и патентообладатель ГОУ ВПО Курский государственный технический университет. № 2007117962/09; заявл. 14.05.2007; опубл. 20.11.2008, БИМП № 17.
3. **Морозов В., Тарахтелюк А.** Диагностика и ремонт НЖМД типа Винчестер. М.: Звезды и С, 1993. 103 с.
4. **Касперски К.** Восстановление данных. Практическое руководство: Пер. с англ. СПб.: БХВ-Петербург, 2006. 352 с.
5. **Воройский Ф. С.** Информатика. Новый систематизированный словарь-справочник (Вводный курс по информатике и вычислительной технике в терминах). 2-е изд., перераб. и доп. М.: Либерия, 2001. С. 536.
6. **Кэрриэ Б.** Криминалистический анализ файловых систем. СПб.: Питер, 2007. 480 с.
7. **Михайлов Д**. Файловая система NTFS. Центр "Информика". URL: http://www.ixbt.com/storage/ntfs.html (24 июля 2000).
8. **Гук М.** Интерфейсы устройств хранения: ATA, SCSI и другие. Энциклопедия. СПб.: Питер, 2007. 477 с.
9. **Электронная** энциклопедия Wikipedia: "Сравнение операционных систем", 2008. URL: http://en.wikipedia.org/wiki/Comparison_of_operating_systems.

УДК 004.056.5

В. М. Амербаев, д-р техн. наук, гл. науч. сотр.,
Институт проблем проектирования
в микроэлектронике РАН,
А. В. Максименко,
руководитель группы заказных разработок,
ООО "Аргуссофт Интернешнл",
e-mail: wizdaz@mail.ru

Модулярные рюкзачные преобразования в информационных технологиях

Рассматриваются вопросы синтеза модифицированной конгруэнтной рюкзачной криптосистемы, удовлетворяющей условиям Варновского и Костера—Одлыжко для противостояния решеточной атаке и обеспечения вычислительной трудности в среднем решения задачи РЮКЗАК.

Ключевые слова: *информационная безопасность, криптография с открытым ключом, модулярный рюкзак, решеточная криптоатака, сложность в среднем*

Введение

В обзорной работе [1] делается вывод, что будущее рюкзачных криптосистем "туманно". Основой тому служат два обстоятельства:

- первое — фактор единственности решения, обусловленный инъективностью любого криптографического преобразования, который исключает любую криптосистему, формируемую в терминах той или иной NP-полной задачи, из общих рамок NP-полных проблем;
- второе — возникновение двух видов решеточных криптоатак на рюкзачные системы различной формы — LLL-метод, метод целочисленного программирования.

Успех упомянутых решеточных криптоатак можно объяснить исходя из того же фактора единственности — решение уравнения шифрации выделяется из многомерного многообразия возможных его решений тем, что искомое решение оказывается единственным в "фундаментальном гиперкубе", который можно понимать как "окрестность нуля". Тем самым открывается возможность эффективного применения различных методов поиска экстремума.

Постановка задачи

Возникает задача — как исключить слабые стороны синтезируемых рюкзачных криптосистем, сохранив простоту их алгоритмов шифрования.

Возможен множественный подход к решению этой задачи, а именно:

- модификация ключевого уравнения — переход от линейного диофантова уравнения к нелинейному;
- развитие этих подходов на математические объекты более сложной природы, чем числа;
- адаптация ключевого уравнения к особенностям систем связи, например, систем связи звездочного типа;
- использование принципов итеративности и мультипликативности (вместо аддитивности);
- введение "гибкого управления" ключевыми и свободными параметрами уравнения единственности, а тем самым и компонентами открытого ключа;
- увеличение размерности ключевого уравнения, приводящее, в сочетании с перечисленными подходами, к увеличению вычислительной сложности известных криптоатак.

Предыдущие результаты

В работе Н. П. Варновского [2] описан класс математических задач, которые могут быть использованы при решении проблем информационной безопасности, и разработана теория, в рамках которой сформулированы критерии вычислительной трудности в среднем решения задачи РЮКЗАК.

Задача РЮКЗАК по Варновскому. Пусть q_1, ..., q_n — различные простые числа в диапазоне от 2^{b-1} до 2^b и пусть Q — их произведение. Пусть далее a_1, ..., a_m, C — независимые случайные целые числа, распределенные равномерно в диапазоне от 1 до $Q-1$. Задача РЮКЗАК ставится как задача поиска таких значений $x_i = 0, 1$, что

$$\sum_{i=1}^{m} x_i a_i \equiv C(\mathrm{mod}\, Q).$$

Варновский доказал, что если хотя бы одна из задач, описанных Айтаи в работе [3], трудна в худшем случае, то данная задача РЮКЗАК трудна в среднем.

Отметим еще один результат, связанный с трудностью задачи РЮКЗАК, который получили Костер, Жу, Ламаччия, Одлыжко, Шнорр и Стерн [4]. Они формально доказали, что при современной решеточной криптоатаке [5] поддаются эффективному вскрытию рюкзачные криптосистемы с плотностью менее 0,9408. Под плотностью рюкзака понимается величина

$$d = \frac{nr}{\log_2 a_{\max}}, \qquad (1)$$

где n — число элементов рюкзака; r — разрядность элементов шифруемого сообщения; $a_{\max}$ — максимальный размер элемента рюкзака.

ВКО-рюкзак. Таким образом, можно сформулировать *необходимые условия* для синтеза криптографического рюкзака, удовлетворяющего требованиям:

а) противостояние решеточной криптоатаке;

б) обеспечение вычислительной трудности в среднем решения задачи РЮКЗАК.

Комплексом необходимых условий служат:

- конгруэнтная форма рюкзака с равномерно распределенной последовательностью элементов (условие Варновского);
- плотность рюкзака > 0,9408 (условие Костера—Одлыжко);
- высокая размерность n и разрядность элементов не менее 32 бит (условие для противостояния решению задачи РЮКЗАК методом линейного программирования).

Рюкзак, удовлетворяющий перечисленным условиям, назовем ВКО-рюкзаком (Варновского—Костера—Одлыжко).

Заметим, что конгруэнтная форма рюкзака является нелинейным случаем. Действительно, если перейти от рюкзачной системы "в сравнениях" к рюкзачной системе "в равенствах", то получим диофантово уравнение с нелинейной частью:

$$\sum_{i=1}^{n} a_i x_i - \left\lfloor \frac{1}{Q} \sum_{i=1}^{n} a_i x_i \right\rfloor Q = C,$$

где C — наименьший неотрицательный вычет по $\mathrm{mod}\, Q$ суммы $\sum_{i=1}^{n} a_i x_i$.

В статье изучаются принципы синтеза ВКО-рюкзака с управляемыми параметрами ключевого материала.

Введем следующие обозначения:

- n — число элементов рюкзака (размерность);
- b — разрядность элементов ключевого материала;
- r — разрядность элементов сообщения.

Для создания ключевого материала выберем две последовательности простых чисел p_1, ..., p_n и q_1, ..., q_l битностью порядка b, т. е. $2^{b-1} < p_i$, $q_j < 2^b$. Пусть $P = \prod_{i=1}^{n} p_i$ и $Q = \prod_{i=1}^{n} q_j$. В качестве модуля конгруэнтной рюкзачной системы возьмем число Q. В качестве секретных параметров избираются числа p_1, ..., p_n, M (такое, что $(M, Q) = 1$), а также $m_i \in Z^*_{p_i}$, $1 \leqslant i \leqslant n$ (Z^*_p — мультипликативная группа поля Z_p).

Элементы открытого рюкзака $A = (a_1, ..., a_n)$ вычисляются следующим образом:

$$a_i \equiv M a'_i (\mathrm{mod}\, Q) \text{ для } 1 \leqslant i \leqslant n,$$

где $a'_i = m_i P_i (\mathrm{mod}\, Q)$ и P_i имеет вид

$$P_i = \prod_{j=1, j\neq i}^{n} p_j.$$

Числа m_i — специальные параметры, которые подбираются с помощью генератора случайных чисел для достижения равномерного распределения a_i и обеспечения высокой компактности рюкзака.

Алгоритм расшифрования базируется на китайской теореме об остатках [6].

Для обеспечения однозначного расшифрования требуется соблюдение *условия единственности* [7, 8]:

$$b > r,$$
$$n2^r P < Q. \qquad (2)$$

Рассмотрим *процедуру шифрования*. Пусть $X = (x_1, ..., x_n)$ — сообщение, где x_i — r-битные чис-

ла. Тогда, согласно Варновскому, криптограмма C конгруэнтного рюкзака строится по формуле

$$C \equiv \sum_{i=1}^{n} a_i x_i (\mathrm{mod}\, Q) (C \in Z_Q). \quad (3)$$

Процедура расшифрования состоит из последовательности действий

$$1)\ CM^{-1} \equiv \sum_{i=1}^{n} a_i' x_i \equiv C' (\mathrm{mod}\, Q). \quad (4)$$

В силу условия единственности (2) последнее сравнение по $\mathrm{mod}\, Q$ превращается в равенство, что обеспечивает правомерность второго шага расшифрования.

$$2)\ C'(m_i P_i)^{-1} (\mathrm{mod}\, p_i) \equiv m_i P_i x_i (m_i P_i)^{-1} (\mathrm{mod}\, p_i) = x_i \text{ для } 1 \leqslant i \leqslant n.$$

Алгоритм

Для достижения равномерного распределения элементов рассматриваемого ВКО-рюкзака (т. е. выполнения условия Варновского) предлагается следующий алгоритм независимого подбора случайных параметров криптосистемы при заданных n, b.

1. Полагаем, что $n \geqslant 100$, $b \geqslant 32$.

2. Разрядность элементов сообщения r выбираем равной $b - 1$, тем самым соблюдается первое требование условия единственности (2).

3. Выберем случайным образом (без возвращения), в соответствии с законом равномерного распределения, номера произвольно упорядоченных простых чисел p_i из диапазона $(2^{b-1}, 2^b)$ общим количеством равным n и вычислим $P = \prod_{i=1}^{n} p_i$.

4. Аналогичным образом выберем l различных простых чисел q_j из диапазона $(2^{b-1}, 2^b)$ таких, что $\forall i, j,\ p_i \neq q_j$, $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant l$, и вычислим $Q = \prod_{j=1}^{l} q_j$. При этом желательно, чтобы количество l выбираемых простых чисел q_j было минимальным при одновременном соблюдении второго требования условия единственности (2).

5. Выберем случайным образом целое число M из диапазона $(1, Q)$ такое, что $(M, Q) = 1$.

6. Для каждого i, $1 \leqslant i \leqslant n$, выберем случайным образом, в соответствии с законом равномерного распределения, число m_i из диапазона $(0, p_i)$.

7. Вычислим $a_i = m_i P_i M \mathrm{mod}(Q)$.

8. Случайные параметры $p_1, ..., p_n, M, m_1, ..., m_n$ относятся к разряду секретных, параметры $q_1, ..., q_l, a_1, ..., a_n$ — к разряду открытых.

Значения плотности ВКО-рюкзака

n	MIN	MAX	AVG
10	0,814	0,888	0,819
20	0,888	0,934	0,894
30	0,916	0,924	0,920
40	0,932	0,956	0,936
50	0,942	0,947	0,944
60	0,948	0,964	0,951
70	0,953	0,967	0,955
80	0,955	0,970	0,959
90	0,959	0,971	0,961
100	0,961	0,972	0,963

Процедуры всех упомянутых в алгоритме случайных выборок (в соответствии с законом равномерного распределения) реализуются на основе генератора случайных целых чисел, отмасштабированного к диапазону номеров упорядоченной последовательности выбираемых значений параметров. В эксперименте использовались функции для высокоточных вычислений, генераторы случайных и простых чисел из библиотеки NTL [9]. Вычисления проводились на компьютере с процессором 2,4 ГГц Intel Core 2 Duo и 4 Гб ОЗУ.

Далее оценивалась плотность ВКО-рюкзака. Для этого с помощью описанного алгоритма создавалось по 100 случайных ВКО-рюкзаков с параметрами $n = \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$, $b = 32$ и были рассчитаны их плотности по формуле (1). В таблице приведены результаты вычислений распределения максимального (MAX), минимального (MIN) и среднего (ANG) значения плотности ВКО-рюкзака для различных n.

Из приведенной таблицы видно, что при $n \geqslant 50$ и $b = 32$ средняя плотность рассматриваемого ВКО-рюкзака превышает 0,9408 — критический барьер плотности для решеточной атаки.

Заключение

Таким образом, описанный алгоритм способен генерировать конфигурации ВКО-рюкзаков, удовлетворяющие условиям Варновского и Костер—Одлыжко.

Замечание 1. Аналогичный алгоритм синтеза рюкзака, удовлетворяющего необходимым условиям стойкости ВКО-рюкзака, может базироваться на так называемом полиадическом разложении.

Замечание 2. Если обозначить символом $|x|_N$ вычет целого числа x по $\mathrm{mod} N$, то процедура шифрования (3) примет вид матричного модулярного преобразования:

$$\left| \sum_{j=1}^{n} \alpha_{ij} x_j \right|_{q_i} = c_i, \quad 1 \leqslant i \leqslant l, \quad (5)$$

где $\alpha_{ij} = |a_j|_{q_i}$,

а процедура расшифрования (4) — вид:

$$x_j = \|CM^{-1}|_Q \gamma_j|_{p_j}, \quad 1 \leqslant j \leqslant n, \qquad (6)$$

где $\gamma_j = |(m_j P_j)^{-1}|_{p_j}$.

В вычислительном аспекте прямое матричное преобразование (5) представляет собой вычисление по модулю q_i $(1 \leqslant i \leqslant l)$ значений линейной формы (5) с предвычисленными константами α_{ij}, а обратное преобразование (6) требует предварительного формирования величины $|CM^{-1}|_Q$, т. е. восстановления числа по заданной системе остатков

$$c'_i = |c_i \mu_i|_{q_i}, \quad 1 \leqslant i \leqslant l,$$

где $\mu_i = |M^{-1}|_{q_i}$, и последующего вычисления остатков χ_j от восстановленной величины $|CM^{-1}|_Q$ по модулям p_j $(1 \leqslant j \leqslant n)$. Тогда вычисление компонент x_i $(1 \leqslant j \leqslant n)$ сообщения сводится к модульному умножению по модулю p_j остатков χ_j на предвычисленные константы γ_j $(1 \leqslant j \leqslant n)$.

Таким образом, в части трудоемкости вычислений центр тяжести падает на процедуру расшифрации.

Замечание 3. Рассматриваемая криптосистема с открытым ключом может применяться в режиме разового использования для данного сеанса связи при случайной перестановке как секретных оснований $p_1, \ldots, p_n$, так и сгенерированной последовательности $m_1, \ldots, m_n$.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований. Грант № 09-07-00157-а.

Список литературы

1. **Lai M. K.** Knapsack Cryptosystems: The Past and the Future. URL: http://www.ics.uci.edu/~mingl/knapsack.html, 2001.

2. **Варновский Н. П.** Математическая криптография. Несколько этюдов // Московский университет и развитие криптографии в России. Матер. конф. в МГУ 17—18 октября 2002 г. М.: МЦНМО, 2003. С. 98—121.

3. **Ajtai M.** Generating Hard Instances of Lattice Problems // Proceedings 28th Annual ACM Symposium on Theory of Computing. 1996. P. 99—108.

4. **Coster M. J., Joux A., LaMacchia B. A., Odlyzko A. M., Schnorr C. P., Stern J.** Improved Low-Density Subset Sum Algorithms // Computational Complexity 2. 1992. P. 111—128.

5. **Lenstra A. K., Lenstra H. W., Lovász L.** Factoring Polynomials with Rational Coefficients // Matheniatische Annualen. 1982. Vol. 261. P. 513—534.

6. **Виноградов И. М.** Основы теории чисел. М.: Наука, ФМЛ, 1972.

7. **Амербаев В. М., Кожухова Ю. И.** Китайская теорема об остатках в кольце главных идеалов и ранцевая система шифрации // Матер. VII Междунар. семинара // Под ред. Б. М. Лупанова. М.: Изд-во ИПИ, 2001.

8. **Амербаев В. М., Бияшев Р. Г. и др.** Модулярная криптосистема рюкзачного типа // V Междунар. конф. "Рус-Крипто 2003" по современной криптографии и системам защиты информации. М.: 2003.

9. **Shoup V.** NTL: A Library for doing Number Theory. URL: http://shoup.net/ntl.

УДК 519.95; 007:159.955

В. Ю. Лёвин, аспирант,
Московский государственный университет им. М. В. Ломоносова,
e-mail: levval@yandex.ru

Повышение криптостойкости протокола цифровой подписи на эллиптических кривых

Дано подробное описание механизмов повышения криптографической стойкости протокола цифровой подписи Эль-Гамаля на эллиптических кривых. Показано, что данный протокол имеет несколько существенных недостатков. Предложен метод, позволяющий избавиться от этих недостатков, без серьезного изменения стандартного протокола. В результате получена новая схема цифровой подписи, устойчивая к фальсификации подписываемых сообщений как в процессе ее реализации, так и при сеансовой передаче. Изложен метод увеличения стойкости к коллизии стандартных хеш-функций (MD, SHA, RIPEMD, GOST и т. д.).

Ключевые слова: *протокол цифровой подписи, хеш-функции, криптографическая стойкость, эллиптические кривые*

Рассмотрим стандартный протокол Эль-Гамаля цифровой подписи на эллиптических кривых. Фиксируем конечное поле $F = GF(q)$, $q = p^n$, $n \in \mathbb{N}$, p — простое число. Обозначим через $E(F)$ эллиптическую кривую над этим полем. Опишем модификацию протокола цифровой подписи Эль-Гамаля на эллиптической кривой $E(F)$. Пусть точка $P \in E(F)$, $\mathrm{ord}(P) = N$. При этом, согласно протоколу цифровой подписи Эль-Гамаля [1], порядок N точки P эллиптической кривой должен являться достаточно большим простым числом, по порядку близким к $\#(E(F))$, где символ # обозна-

чает мощность множества точек эллиптической кривой $E(F)$. Принимая во внимание теорему Хассе, из которой вытекает единственная асимптотически точная формула, оценивающая порядок группы точек произвольной эллиптической кривой, можно показать, что $N \leqslant q + 1 + 2\sqrt{q}$. После выбора точки P генерируется долгосрочный ключ k_1 и вычисляется новая точка $Q = k_1 P$ эллиптической кривой $E(F)$. Заметим, что множество точек эллиптической кривой $F(F)$ вместе с бесконечно удаленным элементом образуют абелеву группу относительно введенного специальным образом закона сложения точек эллиптической кривой [2], поэтому точка $Q = k_1 P$ будет снова лежать на эллиптической кривой $E(F)$. Пусть $M \in \{0, 1\}^*$ — сообщение, которое нужно подписать. Положим $h(M)$: $\{0, 1\}^* \to \{0,1\}^\delta$ — однонаправленная хеш-функция [3].

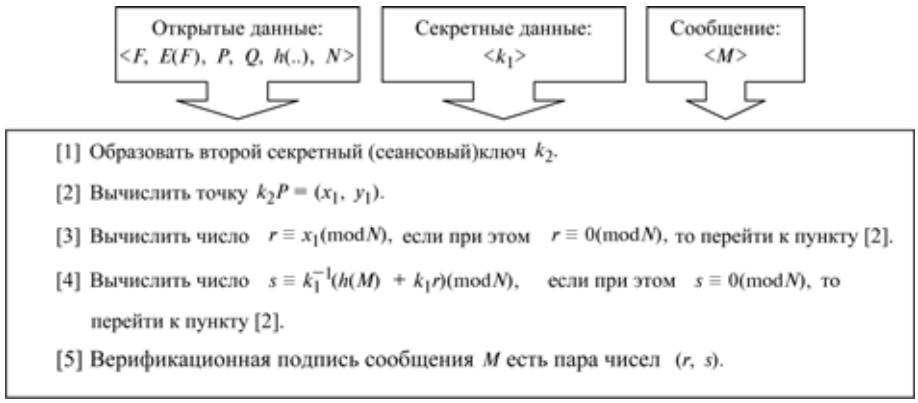

Рис. 1. Генерация цифровой подписи стороной *А*

Рис. 2. Верификация цифровой подписи стороной *Б*

Рассмотрим двух пользователей *А* и *Б*. Пользователю *А* необходимо подписать (генерировать) и послать сообщение $M \in \{0,1\}^*$ пользователю *Б*, а пользователь *Б*, в свою очередь, должен удостовериться (верифицировать цифровую подпись), что сообщение M было послано именно пользователем *А*. Процедура генерации и верификации цифровой подписи Эль-Гамаля, осуществляемая пользователями (далее сторонами) *А* и *Б*, приведена на рис. 1, 2 соответственно.

Покажем что, если $v = r$, то подпись верна. Действительно, $u_1 P = h(M)\omega P = h(M)s^{-1}P$, $u_2 Q = r\omega k_1 P = rs^{-1}k_1 P$. Следовательно, $u_1 P + u_2 Q = s^{-1}P(h(M) + k_1 r) = s^{-1}Pk_2 s = k_2 ss^{-1}P = k_2 P = (x_1, y_1)$, что и требовалось доказать.

Рассмотрим слабые места описанной выше модификации протокола цифровой подписи на эллиптических кривых. Прежде всего следует отметить, что сторона *Б* не обладает данными, достаточными для вскрытия протокола цифровой подписи. Действительно, стороне *Б* нужно знать долгосрочный и краткосрочный (сеансовый) ключи k_1, k_2. Их можно найти, решив задачу дискретного логарифмирования на эллиптической кривой, а именно — зная точки P, Q, найти целое $k_1 : Q = k_1 P$. При этом, зная долгосрочный ключ, искать сеансовый нет смысла, так как он выбирался произвольным образом. Трудоемкость нахождения долгосрочного ключа k_1, согласно [4], носит в общем случае экспоненциальный характер и может быть оценена как $O(\sqrt{q})$. Следовательно, нет быстрых способов найти долгосрочный ключ. Необходимо отметить, что для взлома стандартного протокола цифровой подписи Эль-Гамаля долгосрочный ключ можно и не искать. Действительно, выше уже отмечалось, что значение используемой в протоколе хеш-функции целесообразнее брать близким по порядку к $q + 1 - 2\sqrt{q}$. Вместе с тем, большинство известных на настоящее время хеш-функций имеет фиксированный размер хеш-значений. Для полноты приведем табл. 1 с размерами хеш-значений.

Существует большое число атак на хеш-функции, основанных на парадоксе дней рождений, дифференциальном анализе и других подобных аспектах. Основной недостаток хеш-функций заключается в том, что размер их значений ограничен. Следовательно, с учетом счетности множества всех текстов над $\{0,1\}^*$, обязательно будут существовать коллизии. Это факт означает, что всегда для любой хеш-функции можно предъявить два различных сообщения M_1, M_2 таких, что $h(M_1) = h(M_2)$. Вопрос нахождения коллизий в хеш-функциях является открытым, и, по сути,

Таблица 1

Хеш-функция	Размер хеш-значения, бит
MD2, MD4, MD5	128
RIPEMD-1	128
RIPEMD-2	160
TIGER	192
GOST	254
SHA-1	160
SHA-2	256, 384, 512
HAVAL3	128, 160, 192, 224, 256
HAVAL4	128, 160, 192, 224, 256
HAVAL5	128, 160, 192, 224, 256

Таблица 2

Сообщение (hex-формат)	MD4 — хеш-значение (hex-формат)
4d7a9c83 56cb927a b9d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dd8e31 97e31fe5 2794bf08 b9e8c3e9	5f5c1a0d 71b36046 1b5435da 9b0d807a
4d7a9c83 d6cb927a 29d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dc8e31 97e31fe5 2794bf08 b9e8c3e9	
4d7a9c83 56cb927a b9d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dd8e31 97e31fe5 f713c240 a7b8cf69	e0f76122 c429c56c ebb5e256 b809793
4d7a9c83 d6cb927a 29d5a578 57a7a5ee de748a3c dcc366b3 b683a020 3b2a5d9f c69d71b3 f9e99198 d79f805e a63bb2e8 45dc8e31 97e31fe5 f713c240 a7b8cf69	

ответ на него — дело времени [5]. Например, серия алгоритмов MD была создана в 1990-е годы профессором Массачусетского технологического института (MIT, Massachusetts Institute of Technlogy) Рональдом Райвестом (Ronald Rivest) в целях создания цифровых подписей. Алгоритм MD5 предназначен для использования на 32-битных машинах и является более безопасным, нежели алгоритм MD4, который был взломан. Пример коллизий в MD4 [3] приведен в табл. 2.

Все упомянутые в табл. 1 хеш-функции являются односторонними хеш-функциями, т. е., зная результат преобразования (*message digest*), невозможно восстановить исходную информацию. Таким образом, алгоритмы хеш-функции преобразуют исходную информацию (цифровое сообщение) в число фиксированной длины, называемое "дайджестом сообщения" (*message digest*). Как следствие — неизбежность коллизий. В 1994 г. была обнаружена коллизия в MD5, которая свидетельствует о том, что использование этого алгоритма при определенных условиях ненадежно [3].

Заметим, что при отсутствии у злоумышленника данных о внутреннем устройстве хеш-алгоритма самой "быстрой" атакой является атака, основанная на парадоксе дней рождений. Суть данной атаки заключается в выработке двух серий, состоящих из r_1 и r_2 случайным образом сгенерированных сообщений и выработанных соответствующих хеш-меток. Обозначим через P — вероятность коллизии, а через n — размер хеш-значения в битах. Тогда $P \approx 1 - \mathbf{e}^{-\frac{r_1 r_2}{2^n}}$. При $r_1 = r_2 = 2^{\frac{n}{2}}$ данная вероятность максимальна и равна 0,63. Следовательно, трудоемкость взлома хеш-функции $O(2^{\frac{n}{2}})$. Положив $n = [\log_2 q]$, получаем трудоемкость взлома хеш-функции порядка $O(\sqrt{q})$. Таким образом, в протоколе цифровой подписи Эль-Гамаля вскрытие хеш-функции может является слабым звеном. Действительно, злоумышленник может заранее заготовить два сообщения $M_1 \neq M_2$ таких, что $h(M_1) = h(M_2)$. При этом одно сообщение M_1 он отдаст на подпись стороне *A*. После процедуры подписывания сообщения M_1 злоумышленник отправит свое сообщение M_2. Заметим, что хеш-значения данных сообщений совпадали по построению, значит, процедура верификации ложного сообщения согласно стандартному протоколу Эль-Гамаля пройдет успешно. Предлагается внести следующие изменения в протокол Эль-Гамаля (рис 3, 4).

Здесь $\pi_i(M)$, $i = 1, ..., l$, — текст, полученный случайной перестановкой бит из первоначального сообщения; $\|$ — символ конкатенации; $\tilde{M} = r\|$, $\tilde{M}_l = \pi_i(r\|M_i)$, $i = 1, ..., l$. Следует заметить, что процесс верификации останется без особых изменений.

Предложенная схема усиления протокола цифровой подписи Эль-Гамаля имеет ряд преиму-

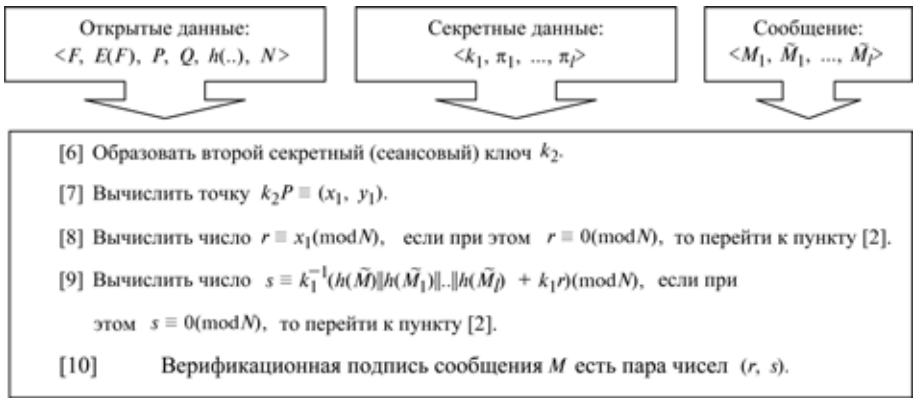

Рис. 3. Усовершенствованная процедура генерации цифровой подписи стороной *A*

Рис. 4. Усовершенствованная процедура верификации цифровой подписи стороной *Б*

ществ. Во-первых, невозможен подлог сообщения, так как злоумышленник не знает заранее, каким будет число r; данное число всегда случайное, так как оно зависит от сеансового ключа. В силу того, что все хеш-функции обладают хорошим лавинообразным эффектом (изменение незначительного числа битов сообщения приводит к серьезному изменению хеш-значения), злоумышленнику неизвестно первоначальное хеш-значение $h(\tilde{M})$. Примеры лавинообразного эффекта приведены в табл. 3.

Во-вторых, сведен к минимуму риск изменения (подлога) сообщения в процессе передачи. Дело

Таблица 3

Название хеш-функции	Значение хеш-функции	Аргумет хеш-функции
MD4	d4fc20d2 d8eb8cc1 c8f45de6 e67b3127	msu
MD4	e766fe3a 2a3b21f3 0fad7361 5329a476	mus
MD5	d62ec24d 065e424d d816ce78 28f62584	mus
MD5	a87ddbe9 f55d9617 142aed3d 7b0f7bd6	msu
RIPEND-1	fefdc2a0 d1aec3c6 a68c33c3 2a5b04f8	msu
RIPEND-1	cb062531 84bd9bec 12c2abb1 9af51d13	mus
SHA-1	9da12f81 2c7925bb f703d5b c5b637dd	msu
SHA-1	1f4196b7 0d242576 4cfb5c6c 1f7752bc	mus
TIGER	40849726 0ce26e96 f08c3a7c 4cc96ebd	msu
TIGER	5400b98c bdf9b159 b593dc28 435e14b3	mus

Таблица 4

Название хеш-функции	Размер хеш-значения, бит	Скорость шифрования, Кбайт/с
MD2	128	23
MD4	128	236
MD5	128	174
HAVAL	128	168
SHA-1	160	75
RIPEMD	128	182

в том, что злоумышленник, зная M, $\tilde{M}_1$, ..., $\tilde{M}_l$, не в состоянии определить соответствующие значения перестановок. Более того, сделать это невозможно. Рассмотрим пример: сообщение M = "мама", сообщение $\pi_1(M)$ = "амма", (перестановка 2134). Однако то же самое получится, если взять перестановку (4231) и т. д. Следовательно, провести атаку на построенную новую хеш-функцию затруднительно, так как придется перебирать перестановки.

В-третьих, увеличена стойкость использующейся в протоколе произвольной хеш-функции. Действительно, поскольку конкатенация значений произвольной однонаправленной хеш-функции снова будет однонаправленной хеш-функцией [6], повышена стойкость к коллизиям за счет увеличения размера хеш-функции. Это важное свойство — ведь стойкость хеш-функции должна соответствовать стойкости всей схемы Эль-Гамаля, а использование хеш-функций фиксированной длины (128,256 и т. д.) является серьезной угрозой стойкости всей системы. Например, используя MD5, коллизию можно построить за 2^{34} операций [3]. Это серьезный недостаток, так как многие используют протокол Эль-Гамаля не с SHA-1, согласно [1, 7], а с другими хеш-функциями, например с MD5. Следует заметить, что первоначальный подлог невозможен, так как злоумышленнику заранее неизвестно число r и вид случайных перестановок. Без использования перестановок сохраняется возможность взлома хеш-функции при сеансовой передаче. Действительно, сама цифровая подпись (r, s) передается по открытому каналу, поэтому злоумышленник может образовать сообщение $\tilde{M} = r \| M$, и далее — вопрос взлома всего протокола — это вопрос взлома соответствующей хеш-функции. За счет увеличения криптостойкости использующейся в протоколе хеш-функции атака на хеш-функцию равносильна атаке на хеш-функцию, размер которой в l раз больше. Этот факт означает, что общая трудоемкость возрастает в $2^{\frac{l}{2}}$ раз.

Предложенный метод усиления криптографической стойкости хеш-функций (точнее стойкости к коллизиям, что в смысле хеш-функций является синонимичным) целесообразно назвать *методом цепи* (или методом перестановок). Метод цепи подразумевает эффективное распараллеливание, что позволяет использовать в различных задачах такие быстрые хеш-функции, как MD4,

MD5 (рассчитанные на эффективную работу на 32-разрядных процессорах). Данный метод позволяет использовать хеш-функции на современных мобильных устройствах, обеспечивая на высоком уровне их криптостойкость. Стоит отметить, что увеличение скорости работы приводит к увеличению общей скорости шифрования. Для сравнения приведена табл. 4, где дана характеристика максимального потока (скорость шифрования на i486SX/33 МГц) [6].

Как следует из табл. 4, модифицированная методом цепи хеш-функция MD4 имеет существенно большую скорость шифрования, чем SHA из классической схемы протокола Эль-Гамаля. Кроме того, метод цепи позволяет реанимировать использование ряда хеш-функций, криптостойкость которых считается сомнительной. Следует также отметить, что приведенные в настоящей статье механизмы улучшения алгоритма цифровой подписи Эль-Гамаля будут иметь место и в других алгебраических структурах, таких как мультипликативная группа конечного поля, группа точек эллиптической кривой, якобиан алгебраической кривой.

Список литературы

1. **Menezes A., Jonson D.** The Elliptic Curve Digital Signature Algorithm // Technical Report CORR 99-34, Dept. of C & O, University of Waterloo, Canada, 2000.
2. **Blake Ian F., Seroussi G., Smart N.** Elliptic curves in cryptography, Cambridge University Press, 1999.
3. **Xiaoyun Wang, Dengguo Feng, Xuejia Lai, Hongbo Yu.** Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD, Dept. of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai, China, 2004.
4. **Левин В. Ю., Носов В. А.** Повышение криптостойкости систем при переходе на эллиптические кривые // Современные проблемы математики, механики и их приложений. Матер. Междунар. конф., 2009. 364 с.
5. **Menezes A., Oorschot P., Vanstone S.** Handbook of applied Cryptography. CRC Press, 1996.
6. **Шнайер Б.** Прикладная криптография. 2-е изд. Протоколы, алгоритмы и исходные тексты на языке С. 2002.
7. **Working** Draft AMERICAN NATIONAL STANDARD X9.62-1998 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 1998.
8. **ГОСТ Р 34.10—2001.** Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И СИСТЕМЫ СВЯЗИ

УДК 621.39+519.8

В. Б. Савкин, научн. сотр.,
Институт механики МГУ,
e-mail: vsavkin@imec.msu.ru

Имитационное моделирование механизмов справедливого распределения коммуникационных ресурсов компьютерных сетей между пользователями и приложениями

Рассмотрена задача управления качеством обслуживания приложений разных типов при их одновременной работе в компьютерных сетях с пакетной коммутацией (например в сети Интернет). Предложен подход к обеспечению приемлемого качества обслуживания интерактивных приложений в условиях сильной перегрузки каналов связи. Эффективность предложенного подхода обоснована методом имитационного моделирования.

Ключевые слова: *качество обслуживания, управление перегрузкой, имитационное моделирование компьютерных сетей*

Введение

В настоящей работе рассматриваются подходы к решению задачи обеспечения качества обслуживания в IP-сетях, для которой отсутствуют общепринятые универсальные методы решения, а именно, задачи справедливого разделения пропускной способности каналов связи сети.

Традиционным и, как правило, применяемым на практике принципом доставки пакетов в *IP*-сетях является принцип наилучших усилий (*best effort*), согласно которому все пакеты находятся в равных условиях. Использование такого принципа обслуживания означает, что один пользователь или одно приложение может загрузить канал, а негативные последствия (рост задержек, увеличение вероятности потери пакетов) ощущают все пользователи и все приложения в равной степени. Отмеченные последствия имеют место и в том случае, когда выделяются специальные приложения, для которых осуществляется приоритетное обслуживание, либо когда трафик делится на классы приоритетности. В этом случае пропускную способность каналов также необходимо каким-то образом разделить между всеми потребителями, имеющими одинаковый приоритет. С тем, чтобы предотвратить отмеченные выше негативные по-

следствия для всех пользователей, поставщики услуг связи вынуждены использовать каналы в таком режиме, когда средняя загрузка очень далека от максимальной. Этого можно добиться, например, ограничением полосы, доступной пользователям (в частности, подключением пользователей с помощью медленных линии связи), или путем использования экономических рычагов (оплата за услуги связи в зависимости от объема переданных данных).

Таким образом, задачу можно сформулировать как справедливое разделение ресурса, а именно, пропускной способности канала, между потребителями. Под справедливостью в контексте данной задачи понимается как можно меньшая зависимость доли ресурсов, доступных одному пользователю или приложению, от поведения остальных пользователей или приложений. Будем также учитывать гранулярность — точность классификации пакетов по их потребителю. Идеальной, по-видимому, является ситуация, когда каждый пакет соотносится с индивидуальным конечным пользователем и с одной из его задач. В этом случае можно рассчитывать на нормальную работу широкого класса приложений при практически полностью загруженном канале. Такой вывод следует из того факта, что в сети Интернет большую часть пропускной способности каналов связи потребляют приложения, которые наименее чувствительны к временному падению скорости передачи в "часы пик". В первую очередь, к приложениям с данным свойством относятся различные службы передачи файлов. Внедрение механизмов справедливого разделения ресурсов сможет автоматически ограничить полосу, выделяемую таким "жадным" приложениям. Их использование при этом не снизит качество работы интерактивных приложений, тогда как при отсутствии подобных механизмов интерактивные приложения страдают. В следующих разделах приведены теоретические соображения и результаты экспериментов, обосновывающие и иллюстрирующие данный тезис.

Задача справедливого разделения пропускной способности может быть поставлена как для одного канала — "бутылочного горлышка" сети, так и в целом для сети, имеющей достаточно сложную структуру. В подобных сетях "узким местом" может стать тот или иной конкретный канал в зависимости от характеристик трафика на рассматриваемом небольшом отрезке времени. В первом случае задачу можно решить с помощью современного оборудования, однако с существенными ограничениями, создающими трудности при реализации такого решения на практике. В сети более сложной структуры возможности используемого оборудования приобретают еще большее значение. Как следствие, появляется необходимость поиска новых методов решения сформулированной задачи.

1. Формальная постановка задачи справедливого и эффективного распределения ресурсов

При разработке и использовании механизмов справедливого и эффективного распределения ресурсов интерес представляют такие показатели, как зависимость доли ресурсов, доставшихся данному потоку, от поведения других потоков (показатель справедливости) и средняя загрузка каналов (показатель эффективности). Более детально, справедливое разделение пропускной способности каналов связи означает выполнение следующих условий.

1. Существует гарантированная минимальная доля пропускной способности, на которую может претендовать поток. Это условие означает возможность утверждать (при достаточной интенсивности потока на входе в сеть), что заранее определенный объем данных успешно пройдет через сеть и будет получен на выходе. Интерес при этом представляют также такие параметры, как максимальные и средние задержки, наблюдаемые при условии выполнения некоторых ограничений на интенсивность потока.

2. Существует алгоритм разделения неиспользованной каждым потоком доли пропускной способности между остальными потоками, например, пропорционально некоторым весам, определенным для всех потоков.

Заметим, что наблюдаемые пользователями скорость и другие параметры передачи зависят от транспортных протоколов и протоколов более высокого уровня, используемых на оконечных узлах. В частности, для *IP*-сетей очень существенным фактором является поведение алгоритма управления потоком протокола TCP.

Задача разделения полосы пропускания одного канала между конечным числом потоков, связанных с пользователями или приложениями, чьи требования заранее известны, является довольно хорошо исследованной. Для ее решения разработаны специальные дисциплины очередей, а именно Class-Based Queueing. Алгоритмы для таких очередей предложены, например, в работах [1] и [2]. Кроме того, получены оценки для задержек, создаваемых очередью, например, в работе [3]. Существуют как свободные программные реализации подобных алгоритмов, так и реализации в некоторых моделях сетевого оборудования ведущих производителей. В настоящей работе не ставится цель четко сформулировать и исследовать свойства этих алгоритмов, это отдельная и достаточно сложная задача.

Отдельный интерес вызывает задача распределения ресурсов между потоками в случае, когда эти потоки образуют иерархию. Например, пусть рассматривается сеть, имеющая подключение к Интернет и несколько организаций-пользователей ресурсов глобальной сети. Можно выделять потоки, соответствующие трафику из Интернет к каждой из организаций, и делить ресурсы сети между такими потоками. Внутри каждого из этих потоков можно выделить трафик, предназначенный для конкретного индивидуального пользователя. Далее между такими потоками второго уровня можно делить пропускную способность, доступную организации. В каждом из таких потоков можно выделить отдельное соединение, или микропоток. Получается иерархия потоков, отражающая организационно-административную структуру сети.

Задачу эффективного использования пропускной способности каналов связи можно сформулировать следующим образом: какие значения загрузки каналов достижимы при условии сохранения ожидаемого пользователями качества обслуживания? Получение точного ответа на данный вопрос представляет определенные трудности в свете того, что для решения поставленной задачи потребуется некоторая модель поведения и ожиданий пользователей. По этой причине, как правило, ограничиваются эмпирическими оценками. Считается, например, что (без использования специальных методов справедливого распределения) многодневное среднее от загрузки канала не должно превышать 30...50 % от пропускной способности, иначе в "часы пик" качество обслуживания недопустимо падает. Следует ожидать, что внедрение механизмов справедливого деления пропускной способности позволит заметно увеличить этот показатель при одновременном улучшении качества обслуживания, предоставляемого интерактивным приложениям (за счет уменьшения доли ресурсов, потребляемых "жадными" пользователями и приложениями). Для проверки данного тезиса автор использовал имитационное моделирование, результаты которого представлены в следующих разделах.

2. Модель справедливой очереди

Для анализа поведения справедливой очереди использован симулятор сетей ns-2 [4]. Автором реализован простой алгоритм, управляющий работой модели справедливой очереди (называемый также дисциплиной очереди). Моделируемая очередь состоит из подочередей, функционирующих по принципу FIFO. При этом каждый поток попадает на вход своей подочереди. Согласно реализованному алгоритму, подочереди обходятся по кругу (такой класс алгоритмов называется *round-robin*), и из каждой подочереди выбирается порция данных, т. е. несколько пакетов. Переход к следующей подочереди осуществляется в том случае, когда:

- либо в текущей подочереди не осталось пакетов;
- либо из текущей подочереди в данном раунде было суммарно выбрано не менее q байт.

Кроме кванта данных q, к параметрам алгоритма относится ограничение на длину очереди, которое в данной реализации представляет собой максимальное число пакетов P. При превышении этого ограничения алгоритм удаляет пакет из конца подочереди, содержащей в текущий момент времени наибольшее число пакетов.

Алгоритм реализован в виде класса на языке C++ с привязкой к языку Tcl, как того требует программирование модулей для системы ns-2.

Для очереди, работающей по данному алгоритму, получена следующая оценка доли пропускной способности канала, на которую может рассчитывать[1] каждый поток (как это требуется в постановке задачи):

$$r_g \geqslant \frac{qS}{Nq + (N-1)M},$$

где S — пропускная способность канала; N — число активных потоков; M — максимальный размер пакета (MTU). Действительно, для любого выбранного потока за один раунд алгоритм выбирает из соответствующей подочереди не менее q байт данных (рассматривается случай, когда в этой подочереди всегда достаточно много пакетов), а данных из остальных подочередей — не более $(N-1)(q+M)$ байт в сумме.

Рассмотренная дисциплина очереди отличается простотой реализации за счет точности разделения полосы. Вместе с тем, несмотря на эту особенность, данный алгоритм может продемонстрировать отличия даже такого упрощенного варианта справедливой очереди от обычного подхода *best effort*.

3. Описание проведенных имитаций (экспериментов)

Цель проведенного автором имитационного моделирования состояла в сравнении работы справедливой очереди и обыкновенной очереди типа FIFO при условии одновременной активности приложений, относящихся к двум рассмотренным выше классам, а именно, "жадных" и интерактивных приложений. При этом предполага-

[1] При достаточной интенсивности потока на входе в очередь интенсивность потока на выходе, составленного из пакетов, которые не были отброшены очередью и были выбраны алгоритмом *round-robin*, составит не менее r_g.

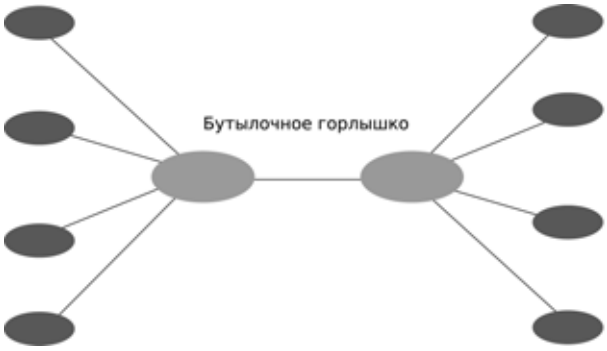

Рис. 1. Модель сети, состоящая из генераторов, приемников и "бутылочного горлышка"

лось, что влияние интерактивных приложений на загрузку канала мало, и основная нагрузка создается "жадными" приложениями. В симуляторе была построена простейшая модель сети типа "бутылочное горлышко", изображенная на рис. 1. Пропускная способность центрального канала — 10 Мбит/с. Значение MTU для тестового трафика было выбрано равным 1480 байт. К сети подключено 100 узлов-генераторов, моделирующих "жадные" приложения, два генератора, моделирующих передачу мультимедийного трафика, и один генератор интерактивного трафика, моделирующий просмотр пользователем веб-страниц. Для каждого генератора по другую сторону центрального канала был подключен соответствующий приемник.

"Жадные" приложения моделировали как передачу больших объемов данных с использованием протокола TCP (вариант Reno). Нагрузку, создаваемую данными приложениями, можно описывать показателем нагрузки

$$\gamma = \frac{N_l B}{S T_E} \cdot 100\ \%,$$

где N_l — число запросов на передачу данных, инициированных в ходе эксперимента; B — объем данных, запланированных к передаче в ответ на каждый запрос[2]; T_E — время эксперимента. Во всех проведенных экспериментах были выбраны значения $T_E = 14\,000$ с (4 ч) и $B = 10^{10}$ байт. Данный показатель игнорирует накладные затраты протоколов передачи файлов, однако позволяет оценить "потребности пользователей" относительно возможностей канала. В проведенных экспериментах показатель нагрузки менялся от низкого уровня (40 %) до состояния крайней перегрузки (> 400 %).

На "бутылочное горлышко" ставилась как очередь типа *drop-tail*, так и справедливая очередь. Кроме того, были использованы две разных модели активности пользователей. В одной из них каждый пользователь мог запускать параллельно несколько *TCP*-соединений, а в другой пользователь ожидал окончания передачи одного файла, прежде чем запустить следующий.

Таким образом, всего было проведено по четыре симуляции (имитационных эксперимента) для каждого значения показателя нагрузки.

В качестве выходных значений симуляций рассматривались показатели качества обслуживания для мультимедийных и интерактивных приложений. Мультимедийные приложения моделировали потоками *UDP*-пакетов постоянной интенсивности, равной 64 кбит/с для одного потока (что меньше гарантированной полосы для данных условий) и 256 кбит/с для другого потока. В качестве показателей качества обслуживания для данных потоков рассматривались доли пакетов, задержка которых при передаче от генератора к приемнику превышала некоторый порог[3]. В проведенных экспериментах были взяты значения порога в 30, 80, 120 и 200 мс, давая вектор из пяти компонент для каждого *UDP*-потока. Качество передачи для некоторого конкретного значения порога можно считать удовлетворительным, если не более 5 % пакетов не уложились в данное значение задержки. Причина выбора указанного допустимого значения доли не доставленных вовремя пакетов состоит в том, что такая ситуация не приводит к ухудшению разборчивости речи при использовании технологии *VoIP*.

Просмотр веб-страницы моделировали как установление *TCP*-соединения и передача 10 000 байт данных. Такая модель игнорирует затраты на посылку *HTTP*-запроса и задержку ответа веб-сервером. Качество обслуживания веб-запросов считалось удовлетворительным, если все передачи завершались не более чем за 30 с. Кроме данного критерия вычислялись медиана и 90-й персентиль эмпирического распределения времени выполнения веб-запроса.

Перед началом моделирования с помощью генератора псевдослучайных чисел были определены моменты времени поступления всех веб-запросов, которые были одинаковыми для всех симуляций, чтобы исключить влияние случайных факторов на результаты сравнения работы очередей. По таким же соображениям для каждого показателя нагрузки были сгенерированы точные моменты поступления запросов на передачу файлов. Данные за первые полчаса (1800 с) каждой симуляции не учитывались при расчете показателей качества обслуживания, так как работа "жад-

[2] Не все запланированные к передаче данные могут быть переданы в ходе эксперимента ввиду ограничения на время.

[3] Отброшенные пакеты считаются имеющими бесконечно большую задержку.

ных" приложений начинается не сразу. По этой причине нужно дать им время для выхода в стационарный режим и достижения запланированной нагрузки.

Для вычисления всех рассмотренных выше показателей на основе протоколов работы симулятора автором были написаны программы на языках *Ocaml* и *Perl.*

4. Результаты моделирования справедливой очереди

Результаты имитационных экспериментов показывают, что используя справедливую очередь, можно добиться достаточно хорошей работы интерактивных приложений, даже в ситуации крайней перегрузки. Так, имитированные веб-запросы обрабатываются удовлетворительно независимо от показателя нагрузки при использовании справедливой очереди, и неудовлетворительно при использовании FIFO в условиях перегрузки. Рост времени обработки веб-запросов в зависимости от нагрузки показан на рис. 2. В двух сценариях, результаты моделирования которых изображены на графиках, использовали две различные модели активности пользователей, которые были описаны в разд. 3.

Высокоскоростное мультимедийное приложение быстро достигает неудовлетворительных показателей при обоих типах очередей. Можно сделать вывод, что для работы подобных приложений в условиях перегрузки нужны другие средства управления качеством обслуживания, например, явное задание приоритета для выделенных приложений.

Показатели качества обслуживания низкоскоростных мультимедийных приложений при росте нагрузки плавно деградируют до определенного

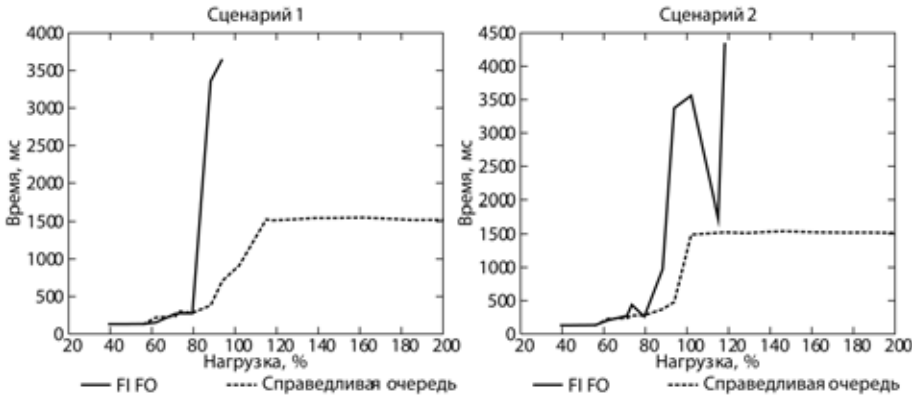

Рис. 2. Время обработки веб-запросов

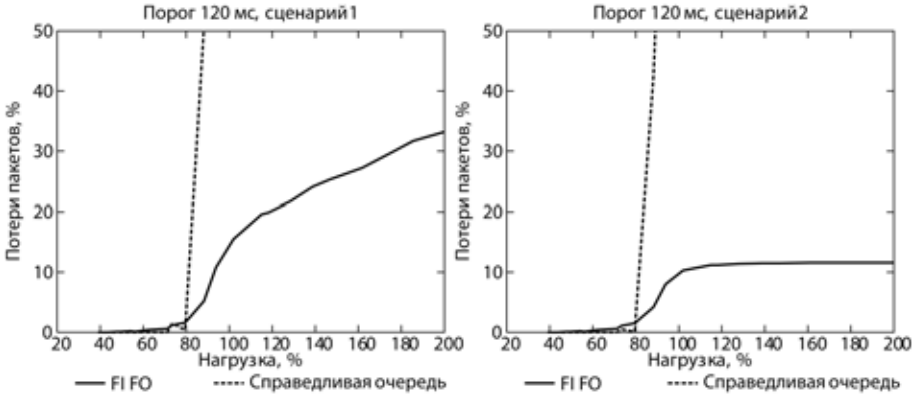

Рис. 3. Показатели качества обслуживания высокоскоростных мультимедийных приложений

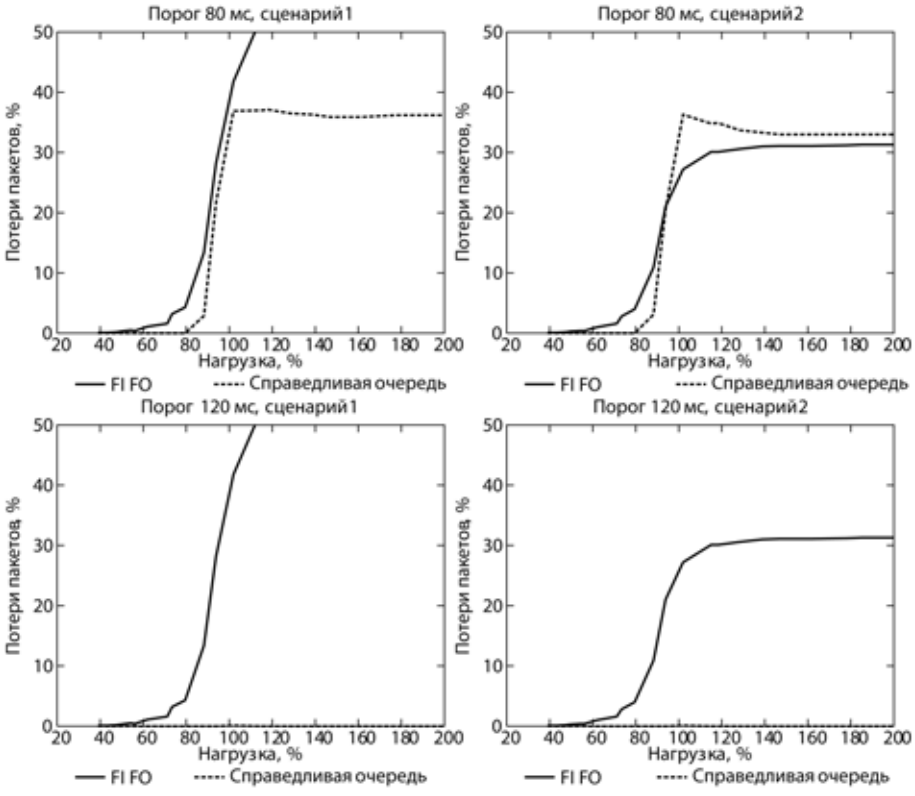

Рис. 4. Показатели качества обслуживания низкоскоростных мультимедийных приложений

предела (в экспериментах наблюдалось отсутствие заметных потерь при пороге 120 мс и наличие существенных потерь при пороге 80 мс) при использовании справедливой очереди и более резко достигают полностью неудовлетворительного уровня при использовании FIFO (уже при нагрузке 80 %). Изменение показателей качества обслуживания мультимедийных приложений с ростом нагрузки показано на рис. 3, 4.

Для обсуждения полученных результатов нужно рассмотреть рост потребностей пользователей со временем. На временны́х промежутках в несколько лет он хорошо приближается экспоненциальным законом [6]. Одновременная резкая деградация в обслуживании всех типов приложений означает внезапное ухудшение оценки уровня сервиса пользователями и, в условиях конкуренции, резкий отток клиентов. Постепенная деградация, начиная с самых требовательных приложений, например высокоскоростных видеотрансляций, позволяет спланировать развитие сети с учетом необходимости расширения каналов связи, установки нового оборудования, внедрения новых механизмов управления трафиком. Изложенные соображения свидетельствуют о наличии потенциального экономического эффекта от внедрения описанного вида механизмов разделения полосы пропускания.

Следует отметить, что в литературе описаны дисциплины очередей с еще более хорошими показателями как в плане справедливости, так и в плане вычислительной эффективности и пригодности для реализации "в железе" [5]. Данная работа предлагает альтернативу общепринятому на текущий момент подходу к развитию сетей путем опережающего расширения каналов с тем, чтобы избежать появления перегрузок. В настоящий момент этот подход представляется наиболее оправданным для сетей магистральных операторов связи, внедряющих и эксплуатирующих решения на основе наиболее популярных и поддерживаемых моделей коммуникационного оборудования. Однако по некоторым экспертным оценкам [6] он перестанет быть применимым по причинам, обусловленным экономическими факторами. Операторы, работающие с конечными пользователями, а также научно-образовательные сети России чаще встречаются с перегрузками. В этой связи для них механизмы управления трафиком в условиях перегрузок еще более актуальны.

На основании изложенных в настоящей работе соображений можно сделать вывод о необходимости разработки и внедрения новых средств управления трафиком в интересах развития как сетей связи общего пользования, так и сетей двойного назначения единой сети электросвязи Российской Федерации.

Список литературы

1. **Floyd S., Jacobson V.** Link-sharing and Resource Management Models for Packet Networks // IEEE/ACM Transactions on Networking. 1995. Vol. 3, no. 4. URL: http://www.cs.utexas.edu/~yzhang/teaching/cs386m-s8/Readings/link-sharing.pdf.

2. **Shreedhar M., Varghese G.** Efficient fair queuing using deficit round-robin // IEEE Transactions on Networking. 1996. Vol. 4, no. 3. P. 375—385.

3. **Kanhere S. S., Sethu H.** On the Latency Bound of Pre-Order Deficit Round Robin // Proc. of the IEEE Conference on Local Computer Networks, Nov. 2002.

4. **The Network** Simulator — ns-2. URL: http://www.isi.edu/nsnam/ns/

5. **Kanhere S. S., Sethu H.** Prioritized Elastic Round Robin: An Efficient and Low-Latency Packet Scheduler with Improved Fairness* Technical Report DU-CS-03-03, Department of Computer Science. Philadelphia: Drexel University. July 2003. PA 19104.

6. **Кипчатов А. А.** Рынок магистрального IP-транзита РФ до 2010 года: тенденции, емкость, цены: доклад на Пиринговом форуме MSK-IX, 2007 г. URL: http://www.msk-ix.ru/download/forum2007/IPtransit_market.ppt

Новая книга

УДК 004.021; 004.75

Н. В. Морев, аспирант,
Владимирский государственный университет,
e-mail: kolyuchiy@gmail.com

Сравнение алгоритмов планирования распределения задач для однородных распределенных вычислительных систем

Проведено экспериментальное сравнение эффективности различных эвристик для решения NP-полной [1] задачи распределения подзадач в однородной распределенной вычислительной системе. Показатель, взятый за основу при сравнении, — длина плана. Исследовано влияние характеристик исходного графа задач и вычислительной системы на эффективность алгоритмов. В качестве базовых характеристик использованы число вершин, число связей и число процессоров.

Ключевые слова: *планирование распределения задач, однородные системы, кластерные системы, распределенные системы, планирование для распределенных систем, алгоритмы планирования, граф задач*

Введение

В наши дни создание высокопроизводительных вычислительных систем немыслимо без использования распределенных вычислительных ресурсов и параллельного выполнения множества задач. При этом разработчики алгоритмов планирования сталкиваются с рядом требований: обеспечение надежности вычислений, обеспечение приоритезации задач, обеспечение заданных уровней обслуживания и т. д. [2]. В статье рассматривается проблема оптимизации распределения задач по узлам по критерию минимального общего времени выполнения всех задач. Сложность этой задачи связана с реализацией двух противоречащих требований: 1) как можно активнее использовать возможности параллелизма; 2) как можно сильнее снизить накладные расходы на пересылку данных между процессорами, на которых выполняются задачи. Так как рассматриваемая задача является *NP*-полной [3], то для ее решения на практике применяют различные эвристики, позволяющие за полиномиальное время получить результат, близкий к оптимальному. Можно выделить несколько классов алгоритмов планирования по используемым подходам: списочное планирование, кластеризация, генетические алгоритмы [3], имитация отжига [4] и др. В статье рассматриваются алгоритмы, относящиеся к первым двум классам, в целях анализа эффективности их применения.

Постановка задачи

Используется модель, основанная на представлении приложения в виде ациклического ориентированного графа задач $G = (V, E, w, c)$, где V — множество вершин-задач; E — связи между задачами, обозначающие какую-либо зависимость (например зависимость по данным), влияющую на порядок выполнения; w — множество весов задач, условно обозначающих вычислительную сложность каждой задачи в виде времени, которое требуется на ее выполнение; c — множество весов связей между задачами, обозначающих накладные расходы, связанные с пересылкой информации между программными модулями, которые выполняются на разных процессорах. Если задачи выполняются на одном процессоре, то вес связи принимают равным 0. Целевую распределенную вычислительную систему представляют в виде множества вычислительных узлов P. Время в задаче задается дискретно, т. е. каждый момент времени может быть обозначен числом из множества Q_0^+ целых положительных чисел, включая 0. Считается, что: 1) каждый процессор в каждый момент времени может обрабатывать не более одной задачи; 2) пересылка информации между процессорами проводится асинхронно с процессом вычисления и не влияет на него. Таким образом, учитывая перечисленные выше условия, задачу можно отнести к типу $P|prec|Cmax$ по предложенной в работе [5] классификации, т. е. к задачам минимизации длины плана с заданными ограничениями порядка выполнения задач на конечном множестве процессоров.

Задача планирования заключается в том, чтобы отобразить множество задач на множество процессоров и определить порядок выполнения задач на каждом из процессоров. Результатом планирования является план, заданный как пара отображений (ts, $proc$), где ts: $V \to Q_0^+$ — функция времени начала выполнения вершин V, а $proc$: $V \to P$ — функция размещения вершин V на множестве P. Таким образом, план описывает временно́е и пространственное размещение задач.

Длину плана S задают как

$$sl(S) = \max_{n \in V} \{tf(n)\},$$

где $tf(n) = ts(n) + w(n)$ — время окончания выполнения задачи.

Исходя из свойств исходного графа задач и вычислительной системы выполнимый план должен удовлетворять двум условиям. Необходимо удо-

стовериться, чтобы ни на одном процессоре в ходе выполнения не оказалось одновременно более одной задачи. Для двух любых вершин $n_i, n_j \in V$

$$proc(n_i) = proc(n_j) \Rightarrow$$

$$\Rightarrow \begin{cases} ts(n_i) < tf(n_i) \leqslant ts(n_j) < tf(n_j) \\ \text{или } ts(n_j) < tf(n_j) \leqslant ts(n_i) < tf(n_i). \end{cases}$$

Для любых $n_i, n_j \in V$, $e_{ij} \in E$, $p \in P$

$$ts(n_j, p) \geqslant tf(e_{ij}, proc(n_i), p),$$

где $ts(n, p)$ — время начала выполнения задачи на процессоре; $tf(e_{ij}, p_i, p_j)$ — время окончания пересылки данных от задачи n_i к задаче n_j с заданными процессорными размещениями p_i, p_j.

План, удовлетворяющий обоим условиям, называется выполнимым.

Очевидно, что для большинства встречающихся на практике графов задач и вычислительных систем существует более одного выполнимого плана. Задача поиска оптимального плана состоит в том, чтобы минимизировать общее время выполнения (длину плана).

Реализация алгоритмов

Для сравнения в среде *MatLab* были реализованы девять алгоритмов планирования, перечисленные в табл. 1, где P, V, E — обозначают соответственно число процессоров целевой вычислительной системы, число задач и число связей между ними. Пять из них относятся к классу алгоритмов списочного планирования, три — к классу алгоритмов кластеризации [3]. Отдельно рассматривается оптимальный алгоритм, так как в силу его большой вычислительной сложности не представляется возможным исследовать его на той же области значений, что и предыдущие алгоритмы.

Списочное планирование заключается в том, чтобы упорядочить вершины графа задач в список задач по какому-либо признаку и затем последовательно по одной задаче формировать план для данной задачи так, чтобы время начала ее выполнения было наименьшим. Порядок задач в списке определяется приоритетом задачи. В качестве приоритета может выступать топологический порядок задач в графе, верхний или нижний уровень вершины графа или их сумма (ВУ + НУ). Нижний уровень вершины в графе — это длина наибольшего пути в графе, начинающегося с данной вершины. Верхний уровень вершины — это длина наибольшего пути в графе, оканчивающегося данной вершиной. В алгоритмах со статическим приоритетом список вершин формируется один раз в начале выполнения алгоритма. В динамических алгоритмах следующая задача, подлежащая добавлению в план, вычисляется на каждой итерации.

Алгоритмы кластеризации задач направлены на минимизацию межпроцессорного взаимодействия за счет объединения сильно связанных между собой задач в кластеры, которые затем отображаются на процессоры заданной вычислительной системы. Здесь объектом рассмотрения на каждой итерации являются не вершины, а ребра графа задач. В алгоритме линейной кластеризации из графа задач сразу выделяют критический путь и формируют из него отдельный кластер, затем для всех оставшихся вершин итерация повторяется. В кластеризации по одному ребру на каждой итерации рассматривается одно ребро и определяется, приведет ли обнуление веса данного ребра к сокращению общей длины плана. Различия между алгоритмами кластеризации по одному ребру могут заключаться в определении порядка, в котором ведется рассмотрение ребер. Результат выполнения алгоритма кластеризации в общем случае еще не является искомым планом для задачи планирования, сначала получившиеся кластеры необходимо отобразить на процессоры целевой вычислительной системы.

Исследуемые алгоритмы планирования

№	Алгоритм	Сокращенное обозначение	Вычислительная сложность
1	Простое списочное планирование (в соответствии с топологическим порядком графа)	ПСП	$O(P(V + E))$
2	Списочное планирование с динамическим приоритетом	СПДП	$O(PV(V + E))$
3	Списочное планирование со статическим приоритетом (верхний уровень)	СПСВУ	$O(P(V + E))$
4	Списочное планирование со статическим приоритетом (нижний уровень)	СПСНУ	$O(P(V + E))$
5	Списочное планирование со статическим приоритетом (ВУ + НУ)	СПСВУ + НУ	$O(P(V + E))$
6	Линейная кластеризация	ЛК	$O(V(V + E))$
7	Кластеризация по одному ребру	КР	$O(E(V + E))$
8	Кластеризация по одному ребру, динамическая	КРД	$O(E(V + E))$
9	Оптимальный алгоритм	ОПТ	NP

Методология проведения эксперимента

Эксперимент проводится в системе *MatLab* версии 7.7.0. Графы задач генерируются случайным образом на основе равномерного распределения следующих характеристик графа: число вершин, число ребер.

Для каждого случайно сгенерированного графа задач выполняются все алгоритмы. Для каждого полученного плана проводится проверка его выполнимости.

Эффективность алгоритма для заданных исходных данных измеряется в виде длины получившегося плана и в виде соотношения длины плана к длине

оптимального плана для выбранного графа там, где длина оптимального плана может быть вычислена.

Характеристики варьируются следующим образом. Число узлов: 1, 10, 100, 500, 1000. Число связей: 0, 1, 10, 100, 500, 1000 (но не больше $(v^2 - v)/2$, где v — число узлов). Число процессоров: 1, 2, 4, 10, 100, 1000 (но не больше числа задач). Выполнено по одному эксперименту для каждого алгоритма при всех указанных сочетаниях параметров.

Экспериментальные результаты

Зависимость длины плана от числа задач

При сравнении зависимости длины плана от числа узлов в графе задачи за основу для сравнения принят алгоритм ПСП. Результаты всех остальных алгоритмов показаны в процентном отношении к основному (рис. 1). При фиксированном числе узлов взято усредненное значение результата выполнения алгоритмов.

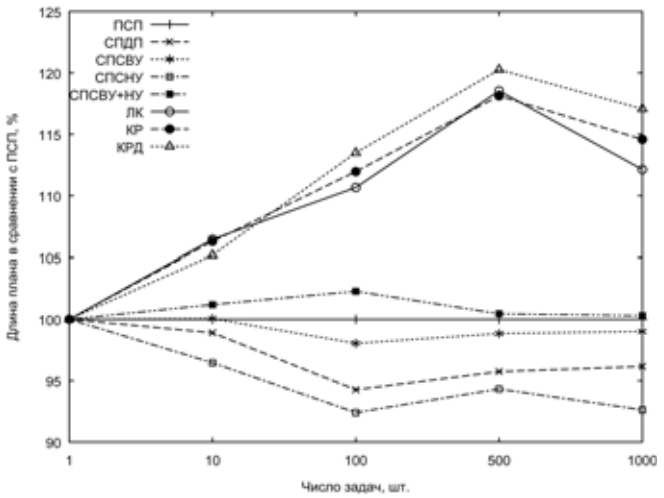

Рис. 1. Зависимость длины плана от числа задач

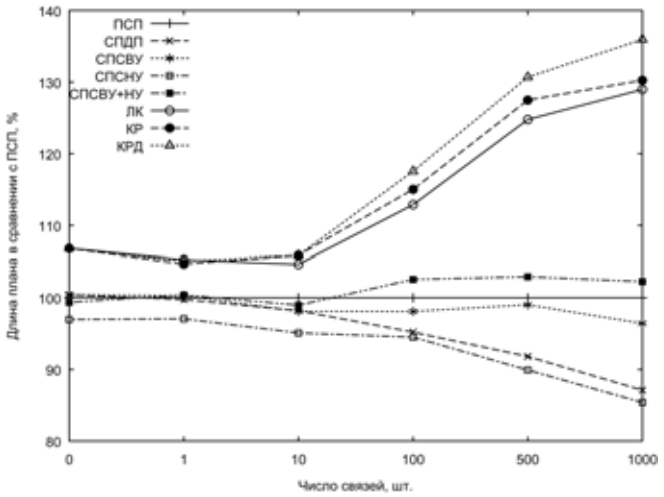

Рис. 2. Зависимость длины плана от числа связей

В вырожденном случае задачи с одной вершиной имеется единственное очевидное и оптимальное решение, поэтому все алгоритмы дают одинаковый результат и все графики выходят из одной точки. Из анализа графика сразу можно отметить, что с увеличением числа задач алгоритмы разделяются на две обособленные группы с различными свойствами. Относительная длина плана, полученного с помощью кластерных алгоритмов, растет, а у списочных алгоритмов уменьшается или остается в пределах 10 % от базового результата.

Рост длины плана кластерных алгоритмов связан с тем, что при увеличении числа задач увеличивается и число кластеров, которые затем необходимо разместить на процессоры вычислительной системы. Если процессоров меньше, чем кластеров, общая длина плана значительно увеличивается (в отдельных опытах в 2,5 раза по сравнению с алгоритмом ПСП).

Результаты алгоритмов списочного планирования слабо зависят от числа задач в графе. Начиная со 100 узлов, их графики выходят на относительно стабильный уровень. Малые отклонения от базового показателя на числе узлов, меньшем 100, объясняются меньшим числом проведенных опытов, а также тем, что для такого малого числа узлов случайно сгенерированные графы слабее отличаются друг от друга, чем при большом числе узлов.

Зависимость длины плана от числа связей

При сравнении зависимости длины плана от числа ребер в графе задачи за основу для сравнения принят алгоритм ПСП (рис. 2). Результаты всех остальных алгоритмов показаны в процентном отношении к основному. При фиксированном числе ребер взято усредненное значение результата выполнения алгоритмов.

Из анализа графиков можно заключить, что при достаточно малом числе ребер в графе задач все алгоритмы показывают слабо отличающуюся от базового алгоритма эффективность. Это связано с тем, что при практически полном отсутствии зависимостей задача упрощается за счет исключения одного из двух противоречащих друг другу требований — требования минимизации затрат на пересылку информации между процессорами.

Алгоритмы кластеризации пытаются минимизировать длину плана за счет уменьшения связей между кластерами задач, однако с ростом числа связей в графе делать это становится сложнее, поэтому длина плана увеличивается.

Среди алгоритмов списочного планирования наилучшие результаты показывают алгоритмы СПДП и СПСНУ. Алгоритм СПСНУ достигает этого за счет выбора в качестве наиболее приоритетных для планирования тех задач, которые от-

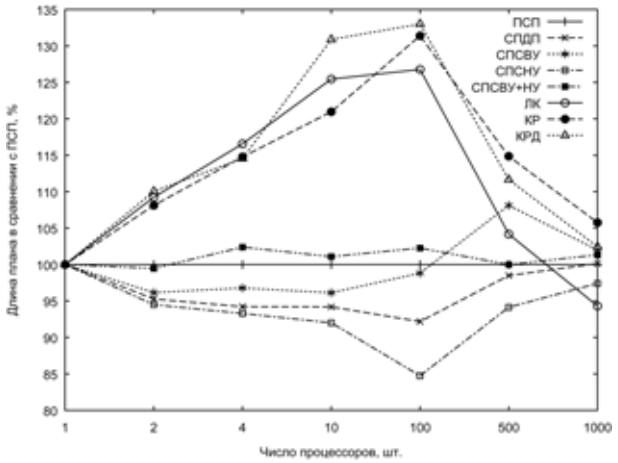

Рис. 3. Зависимость длины плана от числа процессоров

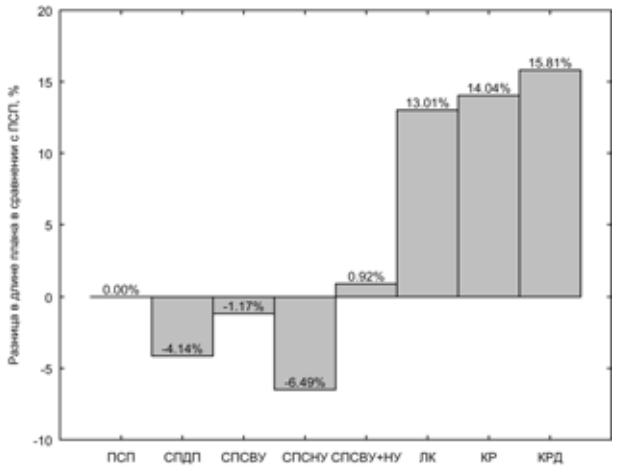

Рис. 4. Сравнение по результатам всех экспериментов

носятся к критическому пути графа. Алгоритм СПДП достигает примерно того же результата путем последовательного перебора всех вершин графа и выбора из них наиболее приоритетной для планирования, что увеличивает сложность алгоритма на фактор, равный числу узлов в графе, и делает его непригодным для практического использования при достаточно большом числе узлов по сравнению с остальными алгоритмами.

Зависимость длины плана от числа процессоров

При сравнении зависимости длины плана от числа процессоров в целевой вычислительной системе за основу для сравнения принят алгоритм ПСП. Результаты всех остальных алгоритмов показаны в процентном отношении к основному (рис. 3). При фиксированном числе процессоров взято усредненное значение результата выполнения алгоритмов.

Анализ полученного графика показывает, что при увеличении числа процессоров целевой вычислительной системы эффективность всех алгоритмов становится примерно одинаковой. Это связано с тем, что с увеличением числа процессоров уменьшается и число проведенных опытов. Кроме того, число задач становится равным или близким к числу процессоров, т. е. практически неограниченным, что значительно упрощает этап отображения задач на процессоры для всех алгоритмов. Для алгоритмов кластеризации это также помогает исключить этап отображения кластеров на процессоры, что положительно влияет на эффективность этих алгоритмов.

Общее сравнение алгоритмов

Общее сравнение рассматриваемых алгоритмов проводилось по усредненным результатам всех проведенных экспериментов. Результаты работы алгоритма ПСП были взяты за основу для сравнения. На рис. 4 показана разница в длине планов, получившихся в результате работы различных алгоритмов.

Выводы

В целом алгоритмы кластеризации показали худшие результаты, чем алгоритмы списочного планирования. Однако из анализа зависимости длины плана от числа процессоров видно, что их эффективность увеличивается с ростом числа процессоров. Это связано с тем, что эти алгоритмы рассчитаны на среду с неограниченным числом процессоров и поэтому требуют дополнительного этапа приведения результата к вычислительной системе с заданным числом процессоров, на котором ухудшается конечный результат.

Алгоритмы списочного планирования показывают примерно одинаковую эффективность. Даже алгоритм СПДП, имеющий большую вычислительную сложность по сравнению с остальными, не дает значительного выигрыша. Среди алгоритмов списочного планирования наилучшие результаты показывает алгоритм с приоритетом узлов по значению нижнего уровня. Среди алгоритмов кластеризации наилучшую эффективность показала стратегия алгоритма ЛК — выделение критического пути в графе задач и отнесение всех его задач к отдельному процессору.

Список литературы

1. **Томас Х. Кормен** и др. Алгоритмы: построение и анализ. Глава 34. NP-полнота. М.: Вильямс, 2006. С. 1296.

2. **Морев Н. В.** Планирование в грид-системах с неопределенными параметрами узлов. // Физика и радиотехника в медицине и экологии: Доклады 8-й Межд. научн.-техн. конф. "ФРЭМЭ-2008". Кн. 1. Владимир: Изд. ВлГУ, 2008. С. 302—305.

3. **Sinnen O**. Task Scheduling for Parallel Systems. New Jersey, USA: John Wiley & Sons — Ho-boken, 2007.

4. **Калашников А. В., Костенко В. А.** Параллельный алгоритм имитации отжига для построения многопроцессорных расписаний // Известия РАН. Теория и системы управления. 2008. № 3. С. 101—110.

5. **Graham R. L., Lawler E. L., Lenstra J. K. et al**. Optimization and approximation in deterministic sequencing and scheduling: a survey // Ann. Discrete Math. 1979. № 5. P. 287—326.

УДК 004.733

А. И. Давыдов, аспирант,
e-mail: davydovai@bk.ru,
В. Г. Шахов, канд. техн. наук, проф.,
И. Б. Ядрышников, аспирант,
e-mail: pr12yad@yandex.ru,
Омский государственный университет
путей сообщения

Анализ абонентской нагрузки в сетях сотовой связи

Проектирование сетей сотовой подвижной связи, как и предварительный расчет параметров базовой станции, обычно начинают с прогнозирования предполагаемой нагрузки, поэтому вопрос о нагрузке в соте и в целом во всей сети является ключевым. Правильный расчет нагрузки делает систему гибкой, готовой к любым неординарным ситуациям. Изучению нагрузки уделяется много внимания, но в большинстве своем это работы по исследованию статистических данных уже работающих сетей.

Ключевые слова: *система массового обслуживания, модель Эрланга, вероятность отказа*

При оценке качества работы сетей сотовой связи (ССС) используется понятие нагрузки. Для реальных ССС различают три вида нагрузок: абонентская, обслуженная и избыточная (рис. 1).

Сотовая сеть является типичным примером системы массового обслуживания (СМО). В ней присутствуют все необходимые характеристики СМО:

- случайный поток заявок;
- продолжительность вызова (длительность занятия радиоканала);
- конечное число обслуживания каналов, предоставляемых подвижным абонентам сотовой сети.

Наибольший интерес с точки зрения СМО представляет модель для расчета абонентской нагрузки в соте с учетом конкретных параметров оборудования базовых станций.

Нагрузка, поступающая на вход системы обслуживания (абонентской нагрузки) от группы источников, есть число поступивших вызовов за время, равное средней длительности одного занятия канала связи. Поскольку нагрузка — величина случайная, то для нее также можно определить математическое ожидание.

Рис. 1. Нагрузки на базовой станции

При оценке абонентской нагрузки пользуются распространенной моделью Эрланга для систем с отказами — вероятность поступления вызова в момент, когда все каналы заняты. Уравнение представляет собой известную формулу Эрланга:

$$p_a = \frac{A^n}{n!} \Big/ \sum_{i=0}^{n} \frac{A^i}{i!}, \qquad (1)$$

где p_a — вероятность отказа; A — абонентская нагрузка, Эрл; n — число каналов.

Представленная формула Эрланга табулирована. Но для практического применения данная формула неудобна. Сложность процедуры определения нагрузки непосредственно с помощью формулы не позволяет рекомендовать ее для инженерного использования, тем более что по ее виду ничего нельзя сказать о характере зависимости допустимой нагрузки от вероятностей отказа и числа каналов.

Решение уравнения относительно нагрузки невозможно, так как искомая переменная входит в состав числового ряда. Однако мы можем воспользоваться одним из приближенных методов, например, с помощью итерационной процедуры Ньютона.

Используя формулу Стирлинга для вычисления факториала

$$n! \approx \sqrt{2\pi n^{n+1}\mathbf{e}^{-n}} \qquad (2)$$

и итерационный метод Ньютона, получим следующее отношение:

$$p_a = \left[\mathbf{e}^{-A}\left(\frac{Ae}{n}\right)^n (2\pi n)\right)^{-\frac{1}{2}}\Big]/f(n), \qquad (3)$$

где

$$f(n) = \mathbf{e}^{-A} \sum_{i=0}^{n} \frac{A^i}{i!}. \qquad (4)$$

Можно воспользоваться приближением:

$$f(n) \approx f(n_1) = \mathbf{e}^{-A} + A\mathbf{e}^{-A}. \qquad (5)$$

Аппроксимируя выражения (5) и (3), получаем следующий приближенный вариант уравнения (1):

$$p_a = \exp\left(-A\left[1 - \mathbf{e}^{\left(\frac{-n\ln(1{,}781A)}{A-0{,}5}\right)}\right]\right)\left(\frac{Ae}{n}\right)^n (2n\pi)^{-\frac{1}{2}}. \qquad (6)$$

Логарифмируя выражение (6), получаем:

$$R = n + n\ln\left(\frac{A}{n}\right) - A + A(1{,}781A)^{-\frac{n}{A-0{,}5}}, \qquad (7)$$

где

$$R = \ln(p_a\sqrt{2\pi n}). \qquad (8)$$

Введем новый параметр $p_{a\text{кр}}$ — критическую вероятность отказа в обслуживании. Он разделяет множество возможных значений p_a на два подмножества:

$$A_1 = \{p_a < p_{a\text{кр}}\},\ \ A_2 = \{p_a > p_{a\text{кр}}\}. \qquad (9)$$

Решая уравнение (7) с учетом выражения (9), получаем:

$$A = \begin{cases} n(1 - z_1),\ p_a \geqslant p_{a\text{кр}}, \\ \dfrac{n}{1 - y_1},\ p_a < p_{a\text{кр}}, \end{cases} \qquad (10)$$

где

$$p_{a\text{кр}} = 0{,}00896 + \frac{\mathbf{e}^F}{\sqrt{2\pi n}}; \qquad (11)$$

$$F = 0{,}561\,\mathbf{e}^{\frac{-(0{,}577 + \ln(n))}{2n}}; \qquad (12)$$

$$z_1 = 2\frac{n - DK}{2nE - 1}; \qquad (13)$$

$$y_1 = \frac{1{,}781D - 1 + \dfrac{E}{2n}}{E - \dfrac{1}{2n} + \dfrac{E}{4n^2}}; \qquad (14)$$

$$D = \ln(p_a\sqrt{2\pi n}); \qquad (15)$$

$$K = (1{,}781n)^{1 + \frac{1}{2n}}; \qquad (16)$$

$$E = \ln(1{,}781n). \qquad (17)$$

График зависимости между поступающей нагрузкой и числом каналов для трех различных значений критической вероятности представлен на рис. 2.

Анализ графиков позволяет сделать следующие выводы:

- определение значения $p_{a\text{кр}}$ является первым этапом оценки допустимого значения нагрузки A и позволяет указать одну из границ диапазона изменения нагрузки;
- исследования расчетов подтвердило целесообразность исследования нагрузки при вероятности потерь от 0,01 до 0,05. Сравнительно небольшое возрастание нагрузки приводит к резкому росту вероятности отказа, т. е. к ухудшению качества обслуживания.

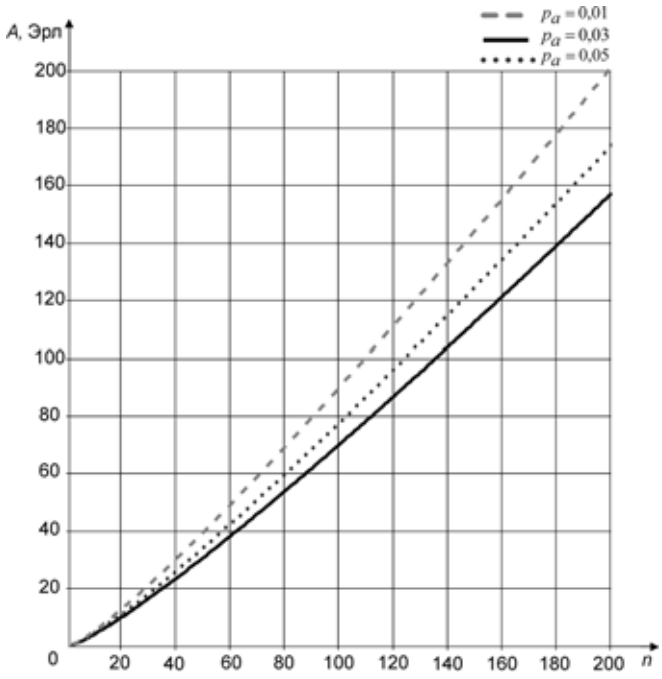

Рис. 2. Зависимость между поступающей нагрузкой и числом каналов

В связи с этим приближенные соотношения, полученные в результате моделирования, представляют собой практический интерес и позволяют определить абонентскую нагрузку с заданной вероятностью отказа при заданном качестве связи.

Вывод: правильный порог вероятности потерь возможно определить только после начала эксплуатации сети, когда нагрузка будет создаваться реальными абонентами с реальным трафиком, но, тем не менее, предварительные расчеты нагрузки позволят заложить тот фундамент, на котором будет основана вся сеть.

Список литературы

1. **Попов В. И.** Основы сотовой связи стандарта GSM. М.: Эко-Трендз, 2005. 296 с.
2. **Ратынский М. В.** Основы сотовой связи. М.: Радио и связь, 1998. 248 с.
3. **Громаков Ю. А.** Технологии определения местоположения в GSM и UMTS. М.: Эко-Трендз, 2005. 143 с.
4. **Берлин А. Н.** Цифровые сотовые системы связи. М.: Эко-Трендз, 2007. 296 с.
5. **Лившиц Б. С.** Теория телетрафика. М.: Связь, 1979. 224 с.

УДК 004.738.5:004.771

Г. А. Тарнавский, д-р физ.-мат. наук, вед. науч. сотр.
Институт вычислительной математики
и математической геофизики СО РАН,
г. Новосибирск,
e-mail: Gennady.Tarnavsky@gmail.com,
С. С. Чесноков, аспирант,
Новосибирский государственный
технический университет,
e-mail: Sergey.Chesnokov@gmail.com

Компьютерное моделирование в Интернете: краткий обзор Web-ресурсов

Проводится краткий обзор современного состояния одного из важных аспектов распространения научных знаний — компьютерного моделирования в Интернете.

Ключевые слова: *информационные технологии, Интернет, компьютерное моделирование, дистанционный доступ*

Введение

Последние достижения в области информационных технологий и Интернета дают основания считать, что в настоящее время уже складываются новые методы организации процесса обмена научными знаниями, а в ближайшем будущем, по-видимому, произойдет повсеместный переход от традиционных способов распространения научных знаний через бумажные журналы и книги к их электронным аналогам. Следует также ожидать массового появления аналогичных электронных форм передачи вещественных результатов научной деятельности в области математического моделирования (вычислительных методов, алгоритмов, компьютерных программ и их финала — полученных числовых значений, структурированных в виде банков табличных и/или графических данных).

Весьма перспективной представляется новая форма передачи знаний в области компьютерного моделирования от автора-разработчика к пользователю-потребителю не через описание алгоритмов и результатов в журналах, а создание возможности непосредственно в любое время ознакомиться с функционированием созданных компьютерных программ и результатами расчета в режиме удаленного доступа через Всемирную сеть, самостоятельно запустив собственную, интересующую его задачу на счет. При этом, как и вообще в науке, так и Интернет-сообществе в частности, передача результатов научной деятельности может осуществляться и безвозмездно, и на платной, коммерческой основе.

В настоящее время компьютерное моделирование используется в интеллектуальной деятельности человечества исключительно широко. Соответственно, в связи с развитием Интернета в нем также широко представлены сайты государственных учреждений, научно-исследовательских институтов, университетов, колледжей, промышленных корпораций, инновационных фирм и других организаций. На этих сайтах содержатся, в той или иной форме, научно-технические продукты, полученные в результате компьютерного моделирования.

Эти сайты можно условно подразделить на три типа:

- представляющие только результаты компьютерного моделирования;
- предназначенные для передачи (продажи) программных комплексов компьютерного моделирования;
- обеспечивающие проведение посетителями (клиентами) непосредственного компьютерного моделирования в Интернете в режиме дистанционного доступа.

Кратко рассмотрим каждый тип этих сайтов.

Сайты, представляющие результаты компьютерного моделирования

Сайты этого типа весьма многочисленны и многогранны, как многогранна вся интеллектуальная деятельность. Поскольку дать им даже краткий обзор представляется затруднительным, приведем примеры только двух таких сайтов.

Первый [1] из них является сайтом Национального управления по аэронавтике и исследованию космического пространства (NASA) США.

На этом сайте содержится обширная информация, отражающая все аспекты деятельности NASA, от истории развития ракетно-космической техники, включая хронологию полетов в космос и описание наиболее важных из них, до изложения различных космогонических теорий, например, теории Большого Взрыва.

На этом сайте приводятся также и некоторые научные результаты, полученные с использованием компьютерного моделирования. В частности, пред-

ставлены данные об атмосфере Марса (рис. 1, см. третью сторону обложки), полученные решением сложной системы уравнений газовой динамики с учетом протекающих в газе физических процессов.

Второй [2] сайт является совместным Web-ресурсом нескольких промышленных корпораций и фирм, выпускающих современные электронные приборы и их компоненты (компьютеры, сенсоры, датчики и т. п.). Эти приборы функционируют на основе наноструктурированных полупроводниковых материалов, состоящих из микроэлектромеханических систем (*microelectromechanical systems,* MEMS). Производство современных MEMS опирается на компьютерную поддержку при проектировании, в частности, микропроцессоров и их основы — наноразмерных транзисторов. Один из типов транзисторов (SON, "*silicon-on-nothing*"), спроектированный с помощью компьютерного моделирования, показан на рис. 2 (см. третью сторону обложки).

Другие сайты этого типа отличаются лишь размерами и контентом, но смысловое содержание является примерно одинаковым, таким же, как на приведенных сайтах [1, 2].

Сайты, предназначенные для передачи (продажи) программных комплексов компьютерного моделирования

Сайты этого типа являются, в основном, специализированными "торговыми площадками" в Интернете, предназначенные для коммерциализации законченных научных разработок, т. е. для продажи или сдачи в аренду программных комплексов математического моделирования в различных областях знания: физике, химии, биологии, медицине и др.

Рассмотрим ряд таких сайтов, ориентированных на области механики сплошных сред [3-5] и микроэлектроники [6—18].

Программные комплексы ANSYS [3], Fluent [4] и FlowVision [5] содержат компьютерные программы решения задач механики твердого тела, аэродинамики и гидродинамики.

Наиболее мощным из них является комплекс ANSYS, который интегрировал более 20 программ с многоцелевыми функциональными возможностями. Комплекс ANSYS коммерчески весьма успешен. Отдельные программы этого комплекса широко используются в различных научных и проектных организациях, в том числе и в России. Так, на рис. 3 (см. третью сторону обложки) показана картина сверхзвукового обтекания объекта с затупленной головной частью и выдвинутой вперед аэродинамической иглой, предназначенной для снижения сопротивления летательного аппарата.

Комплекс Fluent [4] ориентирован на более узкие задачи газовой динамики. Начиная как независимый Web-ресурс, Fluent в настоящее время включен в комплекс ANSYS, как и некоторые другие программы компьютерного расчета задач механики, ранее размещенные на собственных сайтах. Таким образом, сайт ANSYS становится своеобразным программным "гипермаркетом" в Интернете.

Сайт FlowVision представляет отечественные разработки в области газовой динамики и уступает, на наш взгляд, сайтам Fluent и тем более ANSYS.

В микроэлектронике (сайты [6—11]) положение достаточно аналогично. Существует ограниченное число очень мощных торговых брендов, таких как Sentaurus [6] и Pro Suite [9] с их универсальными программными комплексами; узкоспециализированные ресурсы типа CoventorWare [8] и фактически "сошедшие с дистанции" IntellySuite [7], Tsuprem [10] и MicroTec [11], не выдержавшие конкурентной борьбы. Их сайты не обновлялись в течение нескольких лет, а программный комплекс Tsuprem уже интегрирован как раздел в комплекс Sentaurus. Однако, на наш взгляд, программы комплекса MicroTec могут быть, при надлежащей маркетинговой политике, высоко востребованы, поскольку обеспечивают компьютерное моделирование электрофизических, термохимических и механических процессов, применяющихся в нанотехнологиях промышленного производства микроэлектронных устройств.

Так, на рис. 4 (см. четвертую сторону обложки) показано компьютерное моделирование динамики (6 стадий) процессов оксидирования кремниевой подложки Si с созданием слоя диоксида кремния SiO_2, напыления слоя нитрида кремния Si_3N_4 и травления всех слоев Si, SiO_2 и Si_3N_4 для получения требуемых наноструктур в полупроводниковом материале.

Аналогичны и другие Интернет-сайты этого типа. На них размещают: описания комплекса и его разделов, иллюстрации демоверсий компьютерных расчетов, краткие сведения об управлении программами.

Приведем пример еще одного такого сайта. Ресурс www.ELCUT.ru [12] предлагает результаты компьютерного моделирования двумерных электрических полей методом конечных элементов. Навигация по сайту четко организована. Существуют информационные сегменты: "Возможности", "Применение", "Пользователи", "Обучение", "Поддержка", "Наш адрес". Есть демораздел "Бесплатно", демонстрирующий некоторые возможности программно-вычислительного комплекса. Основной раздел, "Купить", содержит систему регистрации посетителей с большим списком вопросов, часть из которых, вообще говоря, не нужна и не вполне уместна. Платежная система организо-

вана по стандартному принципу: вначале нужно перевести деньги через обычную банковскую систему, и Вам вышлют заказанный программный код, который нужно самостоятельно установить и запустить (если удастся, прочитав документацию и переписываясь с разработчиками).

Самым главным здесь является коммерческая информация: стоимость отдельных программ и поддерживающих систем, условия их приобретения и т. п. Клиент должен "приобрести товар" и установить соответствующие вычислительные и операционные системы у себя на компьютере при консультациях специалистов фирм-продавцов.

Сайты, обеспечивающие проведение посетителями (клиентами) непосредственного компьютерного моделирования в Интернете в режиме дистанционного доступа

Обычно передача вычислительного комплекса заключается в приобретении лицензии, документации и кодов компьютерной программы. После этого покупатель устанавливает приобретенный продукт на собственном компьютерном оборудовании.

Как правило, это происходит с большими затруднениями, которые могут быть вызваны разнообразными причинами, от использования разных версий операционной системы до особенностей, установленных у продавца и покупателя поддерживающих систем.

Поэтому в настоящее время появляются новые формы коммерческого (или свободного) использования созданного программного продукта — Центры компьютерного моделирования в Интернете, с кардинальным отличием организации решения задач.

Операции с процессорными системами осуществляются пользователем не на собственном компьютере при установке на нем комплекса, а в режиме дистанционного доступа по Всемирной Сети на его локальном портале — непосредственно в Центре компьютерного моделирования. Это дает возможность посетителю Центра в режиме реального времени провести изучение вычислительного комплекса, организовать решение интересующей его задачи и получить результаты компьютерных исследований.

Такие возможности, пока еще не получившие широкого распространения, предоставляются на сайтах Maple [13] и MatLab [14]. Эти сайты являются большими математическими библиотеками, в которых излагаются практически все разделы математики: алгебра, геометрия, функциональный анализ, дифференциальные уравнения и др. с возможностью их компьютерных иллюстраций. Посетитель, в частности, может провести решение систем линейных уравнений, получить решение какого-либо "не слишком нелинейного" дифференциального уравнения, сделать аналитические выкладки.

Следует, однако, отметить, что в настоящее время провести вычислительные операции непосредственно на сайте MatLab не представляется возможным. Требуются некоторые подготовительные действия, в которых используется включенный в состав комплекса инструментарий MatLab Web Server (MWS), предназначенный для организации удаленного сетевого доступа. Взаимодействие MWS с Web-сервером обеспечивает специальный модуль системы MatLab TCP/IP клиент. Такой подход позволяет размещать сам MatLab и Web-сервер в разных точках сети (подробнее см. [15]).

Главное назначение MWS — инициализация пользователем какого-либо задания: ввод параметров в нужную ему программу MatLab и запуск ее на счет на собственном компьютере (саму программу пользователь менять не может).

При такой организации взаимодействия пользователь избавлен от необходимости устанавливать на свой компьютер всю сложную среду MatLab и изучать особенности ее функционирования.

Так, на рис. 5 (см. четвертую сторону обложки) представлен график некоторой аналитически заданной однопараметрической функции (при шести значениях ее параметра), переведенный в цифровые значения процессорной системой и отрисованный системой визуализации комплекса MatLab. Поэтому, вообще говоря, сайт MatLab пока не является Центром компьютерного моделирования в Интернете в полном смысле этого определения. Это некоторый промежуточный программный инструментарий, поскольку требует от клиента предварительного приобретения, установки на каком-либо сервере (квазиИнтернет-Центре) и лишь затем — организации локальной сети и использования программ MatLab в режиме дистанционного доступа. Очевидно, что следующим этапом развития комплекса станет создание этого "распределяющего" сервера на сайте MatLab с лицензионной или абонентской формой оплаты.

Аналогична ситуация и с инструментарием Maple, который также может быть назван промежуточным этапом между традиционной и новой формой распространения программных продуктов.

В ближайшем будущем следует ожидать массового появления Интернет-площадок, на которых будет осуществляться моделирование предметно-ориентированных больших научных и прикладных задач, связанных с численным интегрированием сложных нелинейных систем уравнений.

Одним из пионеров этого направления является Центр компьютерного моделирования SciShop.ru [16, 17], на котором обеспечивается (в настоящее время) решение задач высокоскоростной аэродинамики, вычислительной астрофизики и проектирования наноструктурированных полупроводниковых материалов.

Подчеркнем еще раз, что в этом Центре проводятся **непосредственные** компьютерные расчеты задач, сформулированных и отправленных на счет самим посетителем сайта, без промежуточной процедуры покупки вычислительных комплексов и установки их процессорных систем у себя на компьютере.

При размещении процессорных систем в Центре не возникают проблемы эксплуатации приобретенных программ. Все интерфейсы налажены, хорошо отработаны и "притерты". В указанных диапазонах вариации параметров функционирование комплексов — безотказное (заметим, что на форуме сайта всегда можно задать любой вопрос и получить разъяснение). Еще одним преимуществом этого подхода является то, что пользователь освобождается от необходимости закупки аппаратного обеспечения (часто весьма недешевого), необходимого для осуществления нужных ему расчетов, фактически "арендуя" его у создателей сайта лишь на время решения своей задачи.

Подчеркнем, что такой эффективный метод эксплуатации процессорных комплексов посетителем Центра потребовал разработки и реализации оригинальных решений. Так как ни один из провайдеров Интернета не позволит проводить на своем узле массовые и, возможно, длительные вычисления, забирающие большие ресурсы и уменьшающие пропускную способность каналов, то потребовалось использовать другую схему проведения расчетов.

В Центре клиент организует вычислительную задачу (выбирает процессорную систему, вводит параметры) и запускает ее на счет. Системы этого сегмента сайта по сопровождению заданий пакуют в файл задание и отправляют его по Сети в вычислительный центр, содержащий спектр компьютеров, в том числе многопроцессорные системы с общей памятью на базе Intel Itanium-2. Здесь выполняется решение задачи, результаты снова пакуются в файл с атрибутами клиента и отправляются либо в Центр, если клиент ждет, либо на его домашний адрес в Сети.

В частности, на рис. 6 (см. четвертую сторону обложки) представлены цифровые значения и графическая визуализация информации (ударные поляры), полученные при компьютерном моделировании ударно-волновых структур течения на входе в воздухозаборник гиперзвукового прямоточного воздушно-реактивного двигателя (ГПВРД).

Заключение

Развитие современных инфотелекоммуникационных систем достигло к настоящему времени той степени, при которой дать даже краткий обзор представляется затруднительным. Роль этих технологий в современном мире не нуждается в специальном обосновании. Исключительное развитие получила, наряду с другими, и торговая деятельность в Интернете с использованием как обычных, так и электронных платежных систем. Широко распространен во Всемирной Сети и обмен результатами научной деятельности как на безвозмездной, так и на платной, коммерческой основе. Ассортимент распространяемой научной продукции очень велик, начиная от статей в научных журналах и кончая пакетами компьютерных программ.

Как и в обычной торговле, здесь бывают свои удачные проекты и неудачные, как лидеры, так и аутсайдеры продаж. Программные комплексы, ориентированные на массового потребителя — от студентов до специалистов, обеспечивающие комфортабельность условий работы, более привлекательны и коммерчески успешны. Парадигмой этого направления развития является появление и совершенствование Центров компьютерного моделирования в Интернете и превращение их в Центры нового поколения — активно посещаемые бизнес-порталы, "особые точки" Интернета по распространению научных знаний.

Список литературы

1. **National** Aeronautics and Space Administration. URL: http://www.nasa.gov.
2. **All** about MEMS. URL: http://allaboutmems.com.
3. **ANSYS,** Inc. — Corporate Homepage (Engineering Simulation for the 21st Century). URL: http://www.ansys.com.
4. **CFD** Flow Modeling Software & Solutions from Fluent. URL: http://www.fluent.com.
5. **Программный** комплекс FlowVision численного моделирования стационарных и нестационарных турбулентных течений жидкости и газа. URL: http://www.flowvision.ru.
6. **ISE** TCAD Sentaurus. Приборно-технологическое моделирование. Система автоматического проектирования URL: http://www.synopsys.com/products/tcad.html (рус: http://www.isen.ru/ise/prod.html).
7. **IntelliSuite:** Industry leading MEMS design tools. URL: http://www.intellisuite.com.
8. **CoventorWare:** The leader in 3D MEMS & semiconductor software. URL: http://www.coventor.com; http://www.cmf.rl.ac.uk/cad/memcad.html.
9. **MEMS** Pro Suite (MEMSCAP): The power of a small world. URL: http://www.memscap.com.
10. **Tsuprem-4:** Semiconductor process simulation software. URL: http://www.synopsys.com/products/tcad/taurus_tsuprem4_ds.html.
11. **MicroTec:** Software Package for Two-Dimensional Process and Device Simulation. URL: http://www.siborg.ca.
12. **Elcut:** электростатика и электродинамика. URL: http://www.elcut.ru.
13. **Maplesoft** — Math Software for Engineers, Educators & Students. URL: http://www.maplesoft.com.
14. **The MathWorks** — MATLAB and Simulink for Technical Computing. URL: http://www.mathworks.com.
15. **Иванов Д. И., Цикин И. А.** Реализация режима удаленного программирования в специализированной среде моделирования MatLab // Информационные технологии. 2008. № 11. С. 23—27.
16. **Тарнавский Г. А., Хакимзянов Г. С., Тарнавский А. Г., Алиев А. В., Малыхин С. М.** Распространение высоких технологий во Всемирной Сети: информационно-вычислительный Интернет-центр компьютерного моделирования научных проблем // Высокие технологии, фундаментальные и прикладные исследования, образование. Т. 5. СПб: Изд-во СПб. Политехнического университета, 2006. С. 83—84.
17. **Центр** компьютерного моделирования. URL: http://www.SciShop.ru.

УДК 004.057.2

Д. В. Силаков, аспирант,
Институт системного программирования РАН,
e-mail: silakov@ispras.ru

Информационно-аналитическая система для разработки и использования базового стандарта операционной системы Linux (LSB)

Предложена логическая модель системы интерфейсов приложений с операционной системой (ОС) Linux и основанный на этой модели метод построения и развития интерфейсных стандартов ОС Linux, нацеленных на обеспечение возможности создания приложений, переносимых между различными дистрибутивами Linux.

Ключевые слова: *переносимость программного обеспечения (ПО), стандарты ПО, Linux*

Проблема разработки переносимых приложений для ОС Linux

В последние годы наблюдается неуклонный рост популярности операционной системы (ОС) Linux, которая все больше используется не только энтузиастами, но и крупными коммерческими компаниями и государственными учреждениями. Тем не менее, доля Linux во многих сегментах рынка (в частности, на настольных компьютерах) все еще мала. Одной из основных причин, препятствующих увеличению доли Linux в таких сегментах, является отсутствие в этой ОС необходимых пользователям приложений. В свою очередь, одной из причин отсутствия приложений является разнообразие существующих ОС, основанных на ядре Linux (и называемых дистрибутивами Linux). Так, по данным [1], существует более 500 дистрибутивов Linux, поддерживаемых в настоящее время, — и это без учета узкоспециализированных систем и систем, создаваемых энтузиастами.

Помимо общего ядра в основе большинства дистрибутивов Linux лежит схожий набор компонентов (библиотек, утилит, средств разработки, прикладных программ), создаваемых сторонними разработчиками. Многие разработчики в мире свободного программного обеспечения (ПО) придерживаются принципа "Выпускайте версии рано, выпускайте версии часто" [2], в соответствии с которым обновления для компонентов могут выпускаться по несколько раз в месяц, что порождает большое число различных версий одного и того же компонента. При этом по функциональности каждая новая версия может отличаться от предыдущей. Кроме того, разработчики дистрибутивов часто модифицируют существующие версии компонентов при добавлении их в свои системы, например, в целях предоставления новой функциональности, которая будет выгодно выделять их продукт среди прочих. В результате функциональность одного и того же компонента в разных дистрибутивах может сильно различаться.

Разнообразие дистрибутивов предоставляет богатый выбор пользователям, однако осложняет создание программ, переносимых между различными системами: разработчикам приложений необходимо удостовериться, что все используемые библиотеки и их интерфейсы присутствуют во всех целевых системах и имеют там одинаковую функциональность.

Используемые разработчиками приложений методы увеличения числа дистрибутивов, в которых может корректно функционировать их продукт, зависят от способа распространения программы. Различают открытое ПО, исходный код которого доступен для просмотра и изменения третьим лицам, а также закрытое ПО, монопольное право на модификацию которого принадлежит правообладателям.

Разработчики приложений с открытым исходным кодом обычно полагаются на разработчиков дистрибутивов, которые включают их программы в свои системы. Как правило, в каждом дистрибутиве есть специалист, ответственный за корректное функционирование конкретного приложения в дистрибутиве, который может адаптировать код приложения в случае необходимости. Однако отметим, что чем больше изменений приходится вносить, тем больше вероятность, что полученная в результате модификаций программа будет сильно отличаться от оригинала и не отражать задумок авторов исходного продукта.

Разработчики проприетарных продуктов, не предоставляющие исходный код, не могут полагаться на производителей ОС и должны обеспечивать совместимость своих продуктов с каждой целевой системой на бинарном уровне — пользователям предоставляются только двоичные файлы программы, и задачей разработчиков приложения является обеспечение корректного функциониро-

вания этих файлов в каждом дистрибутиве. Однако сборка и тестирование приложения на каждом существующем дистрибутиве не являются экономически целесообразными для разработчиков ПО. Поэтому многие производители проприетарного ПО поддерживают лишь ограниченное число дистрибутивов Linux — как правило, в качестве целевых рассматриваются системы, которым принадлежит существенная часть рынка, в частности, SUSE Enterprise Linux и Red Hat Enterprise Linux. Однако конечные пользователи ожидают увидеть продукт "для Linux", а не "для SUSE" или "для Red Hat".

Одним из подходов к облегчению разработки приложений, совместимых с различными дистрибутивами, является стандартизация — выделение требований, которым должны удовлетворять все реализации ОС, соответствующие стандарту. Для разработчиков приложений важна гарантия наличия в системе определенных библиотек и функций.

Число функций-кандидатов на включение в стандарт может быть велико — современный дистрибутив Linux, поставляющийся на одном DVD-диске, содержит несколько тысяч библиотек, экспортирующих сотни тысяч функций. Стандартизация — достаточно дорогой и трудоемкий процесс, включающий в себя, как правило, анализ существующих реализаций, разработку документации и тестов. Поэтому важно уметь оценивать реальные потребности приложений и возможности существующих систем, чтобы в первую очередь включать в стандарт наиболее востребованные элементы.

Ввиду большого числа потенциальных объектов стандартизации размеры стандартов могут оказаться достаточно большими, что может затруднить их использование разработчиками. Облегчить создание ПО, отвечающего требованиям того или иного стандарта, может наличие различного рода инструментов, использование которых в процессе разработки гарантирует соблюдение таких требований. К таким средствам можно отнести тестовый набор, проверяющий приложение на соответствие стандарту. Для поддержки создания приложений, отвечающих требованиям стандарта, может предоставляться специальная среда сборки. Все такие вспомогательные компоненты, образующие окружение стандарта, должны поддерживаться в согласованном состоянии с текстом стандарта.

Важным аспектом стандартизации является разработка профилей — объединение существующих стандартов либо их подмножеств в целях создания спецификации, охватывающей определенный класс целевых систем, каждая отдельная составляющая которых уже рассмотрена в одном из имеющихся стандартов. В области разработки ПО подобные спецификации востребованы, в частности, при создании узкоспециализированных продуктов, например, предназначенных только для работы на сервере либо для работы в мобильных устройствах. Производителям таких приложений, как правило, интересны только ОС, работающие на целевых платформах, и для них полезным является наличие стандарта, описывающего только такой класс систем. При наличии одного или нескольких стандартов, уже охватывающих необходимую область, создание требуемой спецификации может быть существенно упрощено и ускорено за счет использования готовых решений. Однако при выделении подмножеств стандартов, равно как и при их объединении, встает задача обеспечения непротиворечивости и согласованности получаемого профиля.

Одним из наиболее известных интерфейсных стандартов для ОС является Единая спецификация UNIX (Single Unix Specification, SUS), в основе которой лежит стандарт POSIX, созданный для обеспечения переносимости прикладных программ между различными UNIX-подобными системами на уровне исходного кода. При таком подходе стандартизации подлежит Интерфейс программирования приложений (API, Application Programming Interface), основную часть которого составляют функции, предоставляемые заголовочными файлами системы. Гарантируется возможность перекомпиляции приложения из исходного кода в каждой совместимой со стандартом ОС. Альтернативой является стандартизация бинарного интерфейса приложений (Application Binary Interface, ABI), обеспечивающая возможность использования одних и тех же файлов приложения во всех совместимых ОС без необходимости перекомпиляции. Основным объектом стандартизации при этом являются бинарные файлы библиотек, предоставляемые ОС, и экспортируемые ими сущности, называемые бинарными символами. Такой подход в настоящее время используется базовым стандартом Linux (Linux Standard Base, LSB).

Истоки SUS восходят к спецификации Common API Specification, разработанной в начале 1990-х годов альянсом COSE, в состав которого входили все ведущие производители UNIX-систем того времени. Основной целью альянса было выявление интерфейсов, общих для существовавших на тот момент реализаций UNIX. Результирующий список содержал 1170 интерфейсов (по этой причине документ также известен как Spec 1170). В 1992—1993 годах, уже в рамках разработки SUS, было дополнительно исследовано 50 ведущих на тот момент приложений для UNIX и выявлено 130 функций-кандидатов на включение в стандарт [3].

Процесс анализа приложений в ходе разработки SUS и Spec 1170 проводился практически без

автоматизации и сводился к изучению исходного кода аналитиками. В начале 1990-х годов такой подход являлся удовлетворительным и позволил добиться качественного и достаточно полного анализа. Однако к настоящему времени число объектов для анализа, а также их размер существенно увеличились — так, при разработке базового стандарта Linux проводится анализ десятков дистрибутивов и тысяч приложений Linux, рассматриваются сотни библиотек и сотни тысяч интерфейсов. Объем стандарта LSB также существенно превосходит POSIX — последняя версия LSB 4.0 специфицирует около 38 000 интерфейсов. Такие объемы данных приводят к необходимости автоматизации многих процессов, связанных с разработкой и развитием стандарта.

Итак, можно сформулировать следующие задачи, стоящие перед информационно-аналитической системой для обеспечения возможности создания приложений, переносимых между различными дистрибутивами Linux:

- обеспечение проверки приложений и дистрибутивов на совместимость требованиям стандарта;
- планирование развития стандарта;
- создание новых версий стандарта и его профилей.

Метод развития интерфейсных стандартов ОС Linux

Метод развития интерфейсных стандартов ОС Linux, предлагаемый в данной работе, включает несколько этапов.

- Проведение анализа экосистемы Linux:
  - выделение популярных приложений, анализ их требований к библиотекам;
  - сбор информации об основных дистрибутивах.

Состав и свойства приложений и дистрибутивов постоянно меняются за счет выхода новых версий, поэтому необходимо иметь информацию не об одномоментном состоянии экосистемы, а о процессе ее эволюции в течение нескольких последних лет. Должен осуществляться постоянный мониторинг экосистемы Linux, а его результаты, собранные к определенным моментам времени, могут быть использованы для создания очередной версии стандарта, как показано на рис. 1.

- Подготовка новой версии стандарта. Данный этап включает выбор интерфейсов, наиболее востребованных приложениями и предоставляемых дистрибутивами, и построение на основе выделенного списка согласованного замыкания интерфейсов, которые будут включены в стандарт.
- Добавление семантической информации об интерфейсах (в частности, описание предоставляемой функциональности), разработка тестов и внесение необходимых модификаций в систему сертификации на соответствие стандарту.

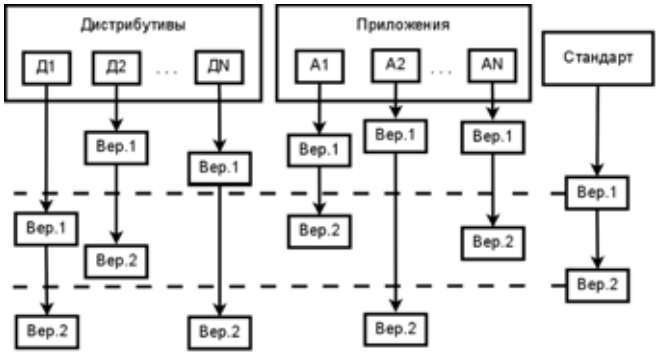

Рис. 1. Анализ экосистемы Linux при разработке стандарта

Предлагаемый процесс опирается на информационную систему, которую можно рассматривать как логическую модель интерфейсов экосистемы Linux. Технологии построения этой модели посвящена основная часть этой статьи.

Логическая модель системы интерфейсов приложений с ОС Linux

В базовом стандарте Linux основным объектом стандартизации является бинарный интерфейс приложений. Поэтому в работе рассматриваются приложения, поставляемые в виде бинарных файлов — исполняемых файлов и разделяемых библиотек. Основным форматом для таких файлов библиотек в ОС Linux является ELF (Executable and Linking Format). Общее описание формата приведено в спецификации System V ABI [4]; дополнения, специфичные для Linux-систем, представлены в соответствующем разделе стандарта LSB [5].

Описываемые стандартом характеристики интерфейсов можно разделить на две группы:

- структурные характеристики, которые могут быть проверены статически, например, состав библиотек или сигнатуры функций;
- семантические характеристики, для проверки которых в общем случае необходимо проведение тестирования, например, поведение функций.

Модель, предлагаемая в данной работе, охватывает структурное описание интерфейсов, абстрагируясь от семантических аспектов. В качестве объектов хранения в модели представлены интерфейсы, задействованные в процессе работы динамического загрузчика системы [6]. Совместимость приложения и дистрибутива по таким интерфейсам позволяет гарантировать успешность запуска приложения в дистрибутиве.

Выделены следующие виды интерфейсов:

- библиотеки — специальный вид файлов формата ELF, которые могут экспортировать интерфейсы;

- бинарные символы, экспортируемые библиотеками, — сущности бинарного уровня, соответствующие функциям и глобальным переменным;
- структура и размер типов, используемых в качестве параметров экспортируемых функций и их возвращаемых значений;
- атрибуты ELF файлов, участвующих в процессе связывания: разрядность, целевая аппаратная архитектура и типы секций, присутствующих в файле.

Предложенная модель не учитывает следующие способы взаимодействия приложений с ОС Linux:

- динамическая загрузка библиотек и обращение к экспортируемым ими символам в процессе работы программы (например, с использованием функций библиотеки "dlopen");
- вызов внешних команд и утилит в процессе работы программы (например, посредством функций "system" или "exec").

Однако в современных рекомендациях по разработке переносимых программ использование подобных возможностей считается приемлемым только в отношении файлов, являющихся частью приложения. Вовлекать же в процесс взаимодействия с ОС любой файл, в наличии которого разработчики не могут быть уверены, считается плохой практикой [7], поскольку может привести к аварийному завершению работы программы.

Информационная система для поддержки развития и использования интерфейсных стандартов ОС Linux

При реализации рассматриваемого метода развития интерфейсных стандартов ОС Linux в данной работе предлагается использовать информационно-аналитическую систему, состоящую из следующих компонентов:

- базы данных, содержащей сведения как о стандартизированных интерфейсах, так и об интерфейсах, используемых существующими приложениями и предоставляемых дистрибутивами. Схема базы основывается на модели системы интерфейсов, рассмотренной ранее;
- инструментов по сбору информации для заполнения базы. Ввиду большого числа существующих продуктов и потенциальных кандидатов на стандартизацию, ручной сбор необходимых данных может оказаться малоэффективным, поэтому необходима автоматизация этого процесса;
- инструментов, использующих базу данных для создания компонентов окружения стандарта.

В базе данных должна храниться информация обо всех интерфейсах и их параметрах, входящих в стандарт и используемых хотя бы одним из компонентов окружения. Если при создании некоторого компонента возникает необходимость узнать список объектов определенного вида, которые включены в стандарт, а также их свойства, установленные стандартом, то эта информация должна извлекаться из базы данных. Такой подход обеспечивает согласованность компонентов относительно стандартизированных объектов. При этом необходимо, чтобы информация в базе была синхронизирована с текстом стандарта. Одним из методов обеспечения такой синхронизации является генерация частей текста с использованием базы данных — в таком случае база становится единственным первичным источником информации о стандартизованных объектах.

Помимо сведений о стандартизованных сущностях для обеспечения согласованности компонентов друг с другом полезно хранить в базе данные, используемые более чем одним компонентом, даже если эти данные не имеют прямого отношения к самому стандарту.

Для хранения информации об элементах существующих дистрибутивов и приложений в данной работе предлагается каждый вид интерфейсов представлять в схеме базы данных двумя сущностями — одна соответствует интерфейсам данного вида, входящим в стандарт, другая — интерфейсам, присутствующим и используемым в реальных системах. Такой подход представляется целесообразным, поскольку данные о стандартизованных объектах и о существующих системах используются с разными целями — для создания компонентов окружения стандарта и для принятия решений — и в общем случае указанные два типа сущностей должны обладать разным набором атрибутов.

Для хранения сведений о нескольких версиях стандарта схема базы должна быть расширена за счет дополнительных атрибутов, содержащих темпоральную (временну́ю) информацию. Существует несколько подходов к введению таких расширений; в данной работе предлагается использовать темпоральную реляционную модель [8], основанную на реляционной модели, но добавляющую каждой сущности дополнительные атрибуты. При использовании такой модели не обязательно наличие специальной темпоральной СУБД — база данных может обслуживаться и реляционными СУБД, являющимися на сегодняшний день наиболее распространенными.

Двумя обязательными атрибутами, добавляемыми темпоральной моделью, являются начальное и конечное время периода жизни сущности — временного интервала, на протяжении которого имеет силу конкретное состояние сущности. Областью значений таких атрибутов в нашем случае являются версии стандарта. Отметим, что темпоральные атрибуты добавляются только к сущно-

стям, соответствующим интерфейсам, входящим в стандарт; для данных о дистрибутивах и приложениях они не требуются.

Программа развития инфраструктуры LSB

Одним из стандартов, специфицирующим интерфейсы ОС Linux, которые должны предоставляться приложениям, является базовый стандарт (Linux Standard Base, LSB). Разработка стандарта ведется международным консорциумом Linux Foundation, объединяющем ведущих участников рынка Linux. Основным наполнением стандарта являются списки библиотек, которые должны присутствовать в системе, и списки бинарных символов, которые должны экспортироваться этими библиотеками. Стандарт активно развивается, включая все большее число элементов — последняя версия (LSB 4.0) специфицирует более 38 000 функций из 57 библиотек, при этом за четыре года, прошедшие с момента выпуска версии 3.0, в стандарт было добавлено более 30 000 функций.

Такое стремительное развитие стандарта выявило ряд проблем в процессе его разработки и в используемой для этого инфраструктуре, в частности, отмечались отсутствие механизма анализа кандидатов на включение в стандарт и трудности в непосредственном использовании текста стандарта, уже в версии 3.0 насчитывавшего несколько тысяч страниц. В 2006 г. в продолжение работ по формализации стандарта LSB [9] стартовала совместная Программа Linux Foundation и Института системного программирования РАН (ИСП РАН) по развитию инфраструктуры стандарта LSB. Целью программы являлось разрешение существовавших на тот момент проблем путем создания информационно-аналитической системы, одновременно способствующей быстрому развитию стандарта и обеспечивающей легкость его использования пользователями — разработчиками дистрибутивов и приложений Linux.

К моменту начала работы программы Linux Foundation в инфраструктуре для разработки стандарта уже присутствовала база данных, содержавшая информацию о стандартизованных элементах. База данных использовалась для генерации частей текста спецификации (списков библиотек, бинарных символов, утилит и т. п.), заголовочных файлов и библиотек-заглушек среды разработки LSB (LSB Development Environment) — среды, призванной облегчить создание приложений, совместимых со стандартом, а также для создания примитивных тестов, проверяющих наличие различных элементов (команд, библиотек, бинарных символов) в системе.

Программа по развитию инфраструктуры LSB в ИСП РАН предполагала выполнение следующих работ:

- разработана схема расширения базы данных для хранения информации о существующих дистрибутивах и приложениях Linux, созданы инструменты для автоматизированного сбора этой информации [10]. В настоящее время в базе содержится информация о 200 дистрибутивах и 1200 приложениях;
- в рамках разработки портала LSB Navigator созданы инструменты для анализа данных о дистрибутивах и приложениях Linux в процессе разработки стандарта LSB в целях выявления кандидатов на стандартизацию, а также проверки ряда формальных требований, предъявляемых к кандидатам на включение в спецификацию;
- разработано темпоральное расширение схемы базы данных, что позволило использовать базу для хранения информации обо всех выпущенных версиях стандарта. В инструменты, работающие с базой, была добавлена возможность создания артефактов, соответствующих любой заданной версии стандарта. Более того, в ряде продуктов, создаваемых на основе базы данных, также имеется поддержка нескольких версий стандарта — так, среда разработки LSB может быть использована для создания приложений, соответствующих любой заданной версии LSB.

В настоящее время ведутся работы по обеспечению поддержки профилей в инфраструктуре LSB, что обусловлено потребностью создания профиля стандарта для мобильных устройств.

Текущая схема окружения базового стандарта Linux приведена на рис. 2.

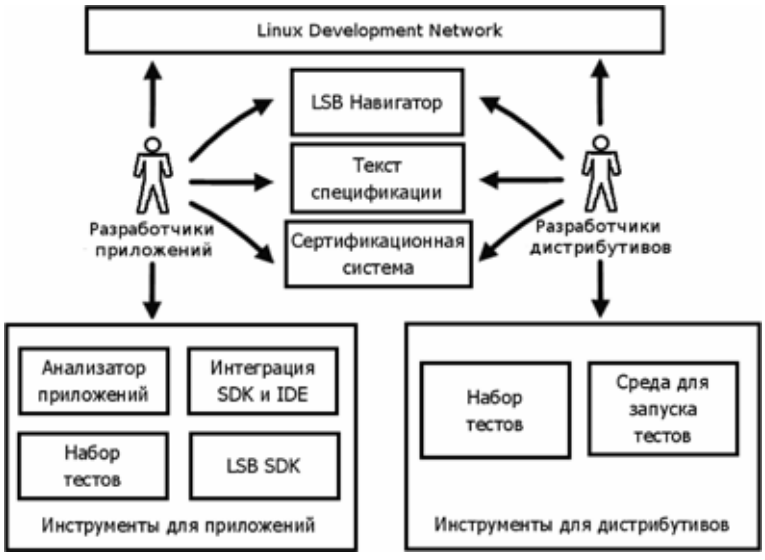

Рис. 2. Схема инфраструктуры базового стандарта Linux

Проект LSB Infrastructure продемонстрировал практическую состоятельность предложенного в данной работе метода развития интерфейсных стандартов ОС Linux. Построенная в ходе проекта инфраструктура базового стандарта Linux позволила автоматизировать анализ существующих дистрибутивов и приложений Linux, существенно ускорив процесс принятия решений по стандартизации тех или иных интерфейсов. Использование единой базы данных для создания различных сопутствующих стандарту артефактов позволяет обеспечивать согласованность таких артефактов с текстом стандарта и между собой. Кроме того, инфраструктура может быть использована для создания профилей базового стандарта Linux.

Список литературы

1. **The LWN.net** Linux Distribution List. URL: http://lwn.net/Distributions/

2. **Raymond E.** The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary // O'Reilly Media, Inc.; Revised & Expanded edition, 2001.

3. **Josey A.** API Standards for Open Systems // The Open Group, 2001. URL: http://www.opengroup.org/austin/papers/wp-apis.txt.

4. **System V.** Application Binary Interface Draft. 24 April, 2001. URL: http://refspecs.freestandards.org/elf/gabi4+/cohtents.html.

5. **Linux** Standard Base Core Specification 3.2. Executable And Linking Format (ELF). URL: http://refspecs.freestandards.org/ LSB 3.2.0/LSB-Core-generic/LSB-Core-generic/elf-generic.html.

6. **Jones M.** Anatomy of Linux dynamic libraries // IBM developerWorks, 2008. URL: http://www.ibm.com/developerworks/linux/library/l-dvnamic-libraries/

7. **Coding** practices for compatibility. // Hewlett-Packard Developer & Solution Partner Program, 1999. URL: http://sysdoc.doors.ch/HP/compat.pdf.

8. **Tansel A. U.** Temporal Relational Data Model // IEEE Transactions on Knowledge and Data Engineering. May—June 1997. Vol. 9. N 3. P. 464—479.

9. **Кулямин В. В., Петренко А. К., Рубанов В. В., Хорошилов А. В.** Формализация интерфейсных стандартов и автоматическое построение тестов соответствия // Информационные технологии. 2007. № 8. С. 2—7.

УДК 004.91

М. В. Петропавловский, д-р техн. наук, проф.,
Д. А. Полевщиков, аспирант,
ФГУ "Национальное аккредитационное агентство в сфере образования", г. Йошкар-Ола,
e-mail: danila.polevshikov@gmail.com

Особенности создания транслятора для языка генерации документов по шаблону в формате WordProcessingML в информационной системе государственной аккредитации

Рассматривается задача генерации документов по шаблону, сведенная к задаче создания транслятора. Для выбранного формата для документов WordProcessingML рассмотрена его структура и особенности применительно к задаче написания транслятора. При практической реализации транслятора была предложена схема деления его на две части — расширенный генератор кода и вызываемый им совмещенный лексический и синтаксический анализатор, разработан алгоритм работы генератора кода и описан способ реализации циклов, задаваемых операторами языка управления генерацией.

Ключевые слова: *трансляторы, генераторы документов, WordProcessingML, шаблоны*

Введение

В практической работе широко используется автоматизированное создание документов с использованием информации из базы данных — генерация документов. Генерация однотипных документов упрощается при использовании шаблонов, содержащих текст, оформление и операторы языка управления генерацией. Такие шаблоны можно рассматривать как исходный код программы, написанный на языке высокого уровня, а сам процесс генерации документа — как трансляцию исходного кода в итоговый документ. Данный класс задач получил название "Обработка шаблонов", или "Трансляция шаблонов"; имеется несколько подходов к их решению.

В общем случае транслятор шаблонов состоит из источника данных, шаблона, непосредственно программы-транслятора и результирующего документа. В зависимости от конкретной области применения существуют различные реализации программы-транслятора. В первую очередь, это трансляторы шаблонов, используемых для разработки веб-сайтов, например smarty для php. Они работают с текстовыми файлами на входе и на выходе, их главная задача — это отделение оформления веб-страницы от бизнес-логики и содержимого.

Кроме того, существует несколько программ для генерации отчетов, также основанных на шаблонах. Генераторы отчетов, такие как Crystal Report или Microsoft Reporting Services, ориентированы на визуальное представление табличной ин-

формации из базы данных в форме тех же таблиц либо диаграмм. При этом используется традиционная компоновка отчета из заголовка страницы, табличных либо графических данных и итогов, и не поддерживается сложное форматирование. Для создания шаблона и просмотра результата обычно используется специализированная программа, возможен экспорт в распространенные форматы.

Также к трансляторам шаблонов можно отнести программы для создания документации на основе исходного кода, например системы классов; программы для генерации исходного кода из абстрактного описания, такого как UML. Для трансляции XML-данных существует специальный язык XSLT со своими практическими реализациями.

Далее рассматривается один из подходов к реализации транслятора шаблонов [1]. Важным отличием предлагаемого подхода от классического "генератора отчетов" является использование стандартной программы Microsoft Word как для создания шаблона вместе со сколь угодно сложным форматированием, так и для работы с итоговым документом. В качестве формата для документов и шаблонов предлагается использовать WordProcessingML. Это основанный на XML формат, который поддерживается Microsoft Word версий 2003 и 2007 и является открытым для сторонних разработчиков.

Транслятор шаблонов в формате WordProcessingML

Документ WordProcessingML состоит из набора узлов верхнего уровня, таких как <w:document Properties>, <w:fonts>, <w:styles>, <w:body> и др. Основное наполнение находится внутри узла <w:body> в трех видах узлов: параграфа <w:p> (paragraph), контейнера для текста с одинаковым форматированием <w:r> (running) и текстового узла <w:t> (text) [2]. Узлы перечислены в порядке возрастания уровня вложенности. Кроме того, метаданные (например, сведения о форматировании или цвете текста) хранятся не в качестве атрибутов, а как дочерние узлы узла, к которому эти метаданные должны применяться, и обычно объединяются в общий узел, например, для элемента w:p создается дочерний узел w:pPr (p properties).

Операторы языка управления генерацией шаблона выделяются в документе с помощью форматирования "скрытый текст", обозначаемого с помощью дочернего узла <w:vanish> у параграфа <w:p > или контейнера <w:r>. При генерации весь скрытый текст удаляется после обработки. Приведем пример шаблона.

Пример 1.

```
<w:p>
<w:r><w:rPr><w:color
w:val = "FF0000"/></w:rPr><w:t>Итого:</w:t></w:r>
<w:r><w:rPr><w:vanish/></w:rPr><w:t>write
(result);</w:t></w:r>
<w:r><w:t>штук</w:t></w:r>
</w:p>
```

Шаблон состоит из одного параграфа и трех текстовых частей: первая содержит простой текст "Итого", форматированный красным цветом; вторая содержит оператор write, выделенный скрытым текстом; последняя — текст "штук".

Рассмотрим особенности транслятора для формата WordProcessingML. Классический лексический анализатор обрабатывает входную последовательность символов и выдает распознанные лексемы, игнорируя комментарии. При этом любая последовательность символов входного потока (лексема), которая согласно грамматике не может быть идентифицирована как лексема языка, обычно рассматривается как специальная лексема-ошибка. Особенностью предлагаемого подхода является то, что статический текст, такой как "Итого:" в примере, не отбрасывается как комментарий, а распознается как неявный оператор вывода, что позволяет избавить шаблоны от явных операторов вывода вида <w:t>write("Итого:"); </w:t> [3].

Следующей особенностью лексического анализатора является необходимость в предварительной подготовке шаблона в связи с тем, что одна лексема, например "write(result);", может быть разделена между несколькими текстовыми частями, перемежаясь с незначащими атрибутами: <w:r><w:t>Wri</w:t></w:r><w:NoProof ><w:r> <w:t>te(result); </w:t></w:r>.

Предварительная обработка заключается в выполнении слияния текстового содержимого таких узлов на основании принадлежности к общему параграфу и схожего форматирования.

Атрибут скрытого текста может быть применен и к параграфу.

Пример 2.

```
<w:p><w:pPr><w:vanish/></w:pPp>
<w:r><w:rPr><w:vanish/></w:rPr><w:t>if(a = b);
</w:t></w:r>
<w:r><w:t>Итого:</w:t></w:r>
</w:p>
<w:p><w:pPr><w:jc w:val = "right"></w:pPp>
<w:r><w:rPr><w:vanish/></w:rPr><w:t>write(result);
</w:t></w:r>
<w:r><w:rPr><w:vanish/></w:rPr><w:t>endif;
</w:t></w:r>
</w:p>
```

Здесь оператор условия if(a = b) находится в первом текстовом блоке первого скрытого параграфа вместе с простым текстом во втором тек-

стовом блоке, а оператор вывода — в следующем за ним обычном параграфе с форматированием по правому краю. Если условие a = b будет истинно, то необходимо будет вывести и слово "Итого:" и обработать оператор вывода write(result), причем скрытый параграф не должен быть выведен в итоговый документ, т. е. слово "Итого:" должно быть перенесено в следующий параграф и отформатированно по правому краю. Если же условие будет ложным, то ничего выводить не нужно.

Данная проблема может быть решена путем использования временного буфера для выходной последовательности и рекурсивного вызова функции обработки входной последовательности. При обнаружении узла параграфа <w:p> выполняется рекурсивный вызов анализатора для обработки его содержимого данного параграфа. При этом для хранения получаемой выходной последовательности используется временный буфер. Если параграф не скрытый и по результату обработки буфер не пуст, то выводится сам параграф и содержимое буфера. Если параграф скрытый, то выполняется переход к обработке следующего параграфа и накапливается буфер, и так до тех, пока не будет встречен обычный параграф и не выведено накопленное содержимое буфера. Следует отметить, что при этом при выводе будет использовано форматирование с последнего не скрытого параграфа.

Алгоритм работы транслятора шаблонов

Опишем алгоритм работы транслятора. Имеются следующие сущности:

- шаблон документа;
- статический текст;
- оформление параграфа;
- оформление текста;
- операторы языка управления генерацией;
- итоговый документ.

Обычный транслятор состоит из лексического анализатора, синтаксического анализатора и генератора кода. При конкретной реализации эти компоненты могут быть разделены или объединены в том или ином виде. Если бы программа на языке управления генерацией представляла собой обычный набор операторов в текстовом файле, то возможным было бы использование традиционной схемы с ведущим синтаксическим анализатором, который вызывает лексический анализатор для получения следующей лексемы и, при достижении определенных состояний, вызывает генератор кода (рис. 1).

В данном случае операторы языка управления генерацией тесно связаны со структурой WordProcessingML-документа и оформлением. Использование неявного оператора вывода для текста шаблона, не отмеченного форматированием "скрытый", и временного буфера для выходной последовательности позволяет применять данную схему. Но при практической реализации возникает неудобство, связанное с необходимостью передавать вместе с лексемой все узлы, обрамляющие текстовый узел, содержащий эту лексему. Поясним на примере 1. Предположим, что был вызван лексический анализатор, который предоставил синтаксическому анализатору лексему оператора "write(result)". Состояние синтаксического анализатора позволяет вывести результат в итоговый документ и вызывать генератор кода, которому необходимо передать не только значение переменной result, но и все узлы форматирования <w:p>, <w:r> и т. д. Следовательно, эти узлы должны быть переданы ранее вместе с лексемой. В случае двух и более текстовых узлов в одном параграфе необходимо отслеживать возможное дублирование узлов параграфа <w:p> при выводе, что не является тривиальной задачей.

Другим, более удобным способом является использование генератора кода как ведущего компонента транслятора, если немного расширить его традиционный функционал (рис. 2).

В предлагаемом подходе фактически необходимо разделить транслятор на две части. Первая часть — расширенный генератор кода — будет транслировать WordProcessingML-шаблон в другой такой же WordProcessingML-документ, выделяя операторы управления генерацией и удаляя либо заменяя скрытый текст. Кроме того, в зависимости от управляющих операторов, например условного оператора if, часть текста шаблона не будет выведена в итоговый документ. Для этого при старте трансляции задаются в true две булевы переменные writeout и writeout_node, разрешающие вывод в итоговый документ соответственно

Рис. 1. Диаграмма последовательности действий для транслятора с ведущим синтаксическим анализатором

Рис. 2. Диаграмма последовательности действий для транслятора с ведущим генератором кода

результата трансляции содержимого текущего узла и самого узла. Вторая переменная writeout_node необходима, поскольку возможна ситуация, когда результат вывести необходимо, но позже, а текущий узел не должен быть выведен. Также по умолчанию устанавливается вывод результата во временный буфер.

Вторая часть транслятора занимается обработкой языка управления генерацией, но не включает в себя генератор кода. Транслятор шаблона, встретив в тексте выделенный форматированием "скрытый" оператор, передает его на вход лексическому анализатору транслятора языка управления генерацией и вызывает его синтаксический анализатор. Результатом его работы является кортеж из переменных, включающий в себя writeout и writeout_node, сообщение об ошибках и, при необходимости, новое содержимое для текущего узла документа. Поскольку вызов транслятора будет осуществляться поэтапно по мере нахождения операторов языка управления генерацией, данный транслятор будет представлять собой интерпретатор.

Алгоритм разбора узла шаблона

Генератор кода осуществляет открытие шаблона с помощью DOM-анализатора и начинает рекурсивный спуск с корневого узла. Чтение шаблона можно осуществлять с помощью любого XML-анализатора, но предпочтительнее использовать DOM-анализаторы, поскольку SAX-анализаторы ввиду своей односторонней направленности затрудняют реализацию циклов или ветвления алгоритма генерации. Кроме того, описанная выше иерархическая структура метаданных также проще обрабатывается с использованием иерархической DOM-модели.

В зависимости от типа текущего узла выполняется одно из следующих действий:

- если узел не подлежит обработке, то пропускаем его;
- если узел — узел параграфа w:p, то выполняем его обработку в зависимости от типа параграфа — скрытый или обычный. Скрытый параграф не может быть выведен в итоговый документ, но результат обработки содержимого сохраняем во временный буфер. Обычный параграф может быть выведен в зависимости от состояния соответствующих переменных. Если параграф выводится и временный буфер не пуст, выводим содержимое буфера;
- если узел — блок скрытого текста, то готовим его для передачи в синтаксический анализатор и вызываем его при накоплении в буфере лексического анализатора одного или нескольких операторов;
- вывод текущего узла в итоговый документ — как есть либо с новым содержимым.

Синтаксический анализатор при вызове получает лексему от лексического анализатора и выполняет синтаксический разбор в соответствии с LL(1)-грамматикой, которая задается набором таблиц [4]. В грамматику включен вызов действий, выполняемых при достижении определенных состояний синтаксического анализатора, таких как присвоение значения переменной, проверка условия и др. При выполнении этих действий могут быть изменены переменные writeout и writeout_node, а также получено новое содержимое для текущего узла, например, выведена переменная.

Наибольший интерес представляет реализация циклов. Если для реализации оператора условия if достаточно изменить переменные writeout/writeout_node, то для циклов необходимо обеспечить повторный вызов участка шаблона в зависимости от результата проверки условия продолжения цикла. Поскольку транслятор предъявляет собой интерпретатор и ведущим является генератор кода, предлагается следующее решение. При вызове синтаксического анализатора дополнительно передается текущий узел шаблона и первоначально пустая переменная loopnode. В начале цикла данный узел запоминается как loopnode и передается обратно генератору кода. Далее осуществляется проверка условия цикла. Если оно положительно, то выполняется обработка тела цикла. При достижении конца тела цикла, если переменная loopnode не пуста, вызывается новый экземпляр генератор кода, начиная от узла loopnode, а предыдущий экземпляр заканчивает свою работу. Поскольку циклы могут быть вложенными, переменная loopnode является

динамическим массивом в форме стека, позволяя запоминать несколько узлов.

Заключение

Рассмотренный транслятор шаблонов в формате WordProcessingML обеспечивает как простоту разработки шаблона и использования получаемых документов, так и возможность расширения функционала, например, за счет добавления возможности генерации графиков. В дополнение к документам в формате WordProcessingML возможно использование основанных на XML форматов для описания графиков: формата VML, совместимого с браузерами, либо формата SpreadSheetML, поддерживаемого Microsoft Office. Некоторые разработки по использованию XML-форматов для графики описаны в [5].

Список литературы

1. **Полевщиков Д. А., Петропавловский М. В.** Язык генерации документов и деловой графики на основе xml-форматов Microsoft Word // Девятые Вавиловские чтения: материалы постоянно действующей всероссийской междисциплинарной научной конференции с международным участием: В 2 ч. Йошкар-Ола: МарГТУ, 2005. Ч. 2. С. 357—359.
2. **Воутер В. В.** OpenXML кратко и доступно. М.: Микрософт Пресс, 2007. 9 с.
3. **Полевщиков Д. А.** Особенности лексического анализатора для языка генерации документов по шаблону в формате Wordprocessingml // Всероссийская молодежная науч. конф. Мавлютовские чтения. Уфа: УГАТУ. 2008. Т. 3. 175 с.
4. **Серебряков В. А., Галочкин М. П.** Основы конструирования компиляторов. М.: Эдиториал, УРСС. 2001. 61 с.
5. **Полевщиков Д. А.** Использование векторного языка разметки (VML) для создания графиков // Современные проблемы фундаментального образования. Матер. VIII региональной научно-метод. конф., посвященной 75-летию Марийского государственного технического университета. Йошкар-Ола: МарГТУ. 2007. С. 31—32.
6. **Ecma** international. TC45 — Office Open XML Formats. Ecma International. URL: http://www.ecma-international.org/memento/TC45.htm.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ И БИОЛОГИИ

УДК 53.082.9:612.172.4 + 51-76:531.761:612.172.2

А. А. Кузнецов, канд. физ.-мат. наук, доц.,
Владимирский государственный университет,
e-mail: artemi-k@mail.ru

Системный анализ и обработка электрокардиографической информации

Каждому кардиоциклу на электрокардиограмме ставится в соответствие реализуемая в его собственном интервале времени потенциальная энергия и топологическая структура комплексов. ЭКГ-информация представлена в форме пяти цифровых рядов абсолютных (RR-интервалов, S_{RR}, R) и относительных (S_R/R, S_R/RR) значений. Метод оценки вариабельности сердечного ритма (ВСР) применен в форме анализа функциональных зависимостей параметров ВСР от стандартного отклонения. Показано, что в такой форме ВСР может служить фактором функционального состояния организма.

Ключевые слова: *электрокардиограмма (ЭКГ), вариабельность сердечного ритма (ВСР), анализ и обработка, функциональное состояние организма (ФСО)*

Постановка задачи

Основным и общепризнанным параметром гомеостаза является кругооборот крови [1]. Системный процесс работы сердца, обладающий функцией самосохранения, стремится удержать параметры системы кровообращения в интервале устойчивости функционального состояния организма (ФСО) [2]. При слабо регулируемом легочном кругообороте и относительно стабильном венозном основными параметрами кровотока, как известно, становятся сердечный выброс и сопротивление сосудистого русла. Сердечный выброс определяется значением ударного объема крови, помноженного на частоту сокращений [1]. Если для донозологической диагностики ФСО параметры сопротивления сосудистого русла назначить регулируемыми посредством комплекса барорецепторов, то исследование ФСО разбивается по двум независимым ортогональным направлениям на электрокардиограмме (ЭКГ): динамика частоты сердечных сокращений (ЧСС, уд/мин) — по оси времени; динамика изменений ударного объема — по оси амплитуд, соответственно. Общее ФСО определяется по результатам системного анализа ЭКГ информации по фазе и амплитуде: по горизонтали — ряд интервалов времени (*RR*, мс) и по вертикали — ряд значений разности потенциалов (φ, мВ).

Целью работы является разработка независимых методов обработки и анализа ЭКГ с созданием информационно-технологической основы системного анализа ЭКГ информации для адекватного определения ФСО.

Материалы и методы

Регистрации ЭКГ проводились на базе лабораторий Владимирского госуниверситета (ВлГУ) у условно здоровых обследуемых (УЗО) и пациентов отделения реанимации областной клинической больницы. Первую группу составляли 32 учащихся ВлГУ обоих полов в возрасте 18—24 года. Для второй группы (юноша К. и девушки Ш. в возрасте 21 год) проводилась серия регистраций ЭКГ в течение 29 и 34 дней. В третьей группе возраст 50 больных с различными заболеваниями варьировался в пределах 30—50 лет. Все измерения с длиной записи около 20 мин проводились монитором Холтера комплекса амбулаторной регистрации электрокардио-сигнала "АппА Flash3000" [3] с программой "EScreen" [4] в режиме покоя с использованием накожных электродов. При регистрации биопотенциалов применялись двухполюсные отведения (по Небу) с расположением электродов, соответствующим переднему грудному отведению (A-*anterior*), или стандартному отведению II с максимальной амплитудой зубцов.

Графики ЭКГ подвергались процедуре дискретизации с шагом 1 мс и сохранялись в форме цифровых рядов в текстовых файлах [4]. Каждый массив из базы данных ЭКГ анализировался с помощью программы определения их координат для идентификации зубцов *R* [5]. Основной технологической проблемой при этом является качественная идентификация пиков зубцов *R*. Применение отведения II при регистрации ЭКГ у здоровых молодых людей снижает актуальность этой проблемы. Алгоритм программы определения координат *R*-зубцов на ЭКГ [5] позволяет составить три массива данных: амплитудных значений *R*, мВ; интервалов времени между соседними зубцами *RR*, мс; площадей кардиоциклов S_{RR}, мВ · с.

Интегральный метод обработки и анализа ЭКГ

Условия генерации и проведения сигнала электропроводящей системой сердца (ЭПСС) предопределяют динамику биопотенциала φ, реализуемую в соответствующую форму кривой ЭКГ [1]. Общая площадь под кривой ЭКГ, ограниченная снизу условной горизонтальной "нулевой" линией, характеризует реализованную потенциальную энергию за период регистрации. Распределение этой энергии в последовательности кардиоциклов (по интервалам времени) позволяет поставить в соответствие каждому кардиоциклу собственный биопотенциал, адекватный его морфологии. Такое представление о работе сердца можно назвать энергетической моделью работы сердца, в рамках которой объектом исследования является цифровой ряд значений последовательности измеренных величин, характеризующих реализуемую потенциальную энергию в интервале каждого кардиоцикла.

В этой части исследуется динамика значения связанной энергии, прямо пропорционального информационной энтропии [2, 6]. Работа ЭПСС может быть представлена в форме разности полной энергии, аккумулируемой проводящей системой, и связанной энергии, определенной топологической структурой кардиоцикла. В рамках этой модели и при ФСО в норме утверждается: 1) связанная энергия — не расходуема; 2) при отсутствии изменений этой энергии и при положительном избытке продукции энтропии все изменения ритма обратимы [2, 6, 7]. Среднее значение площади $<S_{RR},>$, приходящееся на один кардиоцикл, является среднеинтегральной оценкой работы ЭПСС. В таком представлении значение среднего уровня изоэлектрической линии ЭКГ, отложенное от "нулевой линии" (рис. 1), может характеризовать энергетический тонус сердца. Динамика уровня изолинии должна предопределять морфологию комплексов на ЭКГ. Если общий энергетический тонус мал, то полную энергию — общую площадь под ЭКГ достраивает до нормы текущая фаза ростом частоты сокращений, коррекцией амплитуд и ширины отдельных зубцов и т. п. [2, 8—10].

На графике ЭКГ по фиксированным пикам зубцов *R* вертикальными прямыми ограничены *RR*-интервалы. Ограничение снизу задавалось горизонтальной "нулевой" линией, максимально приближенной к кривой ЭКГ и не пересекающей ее (рис. 1, 2, *а*). Выделенные участки кривой ЭКГ

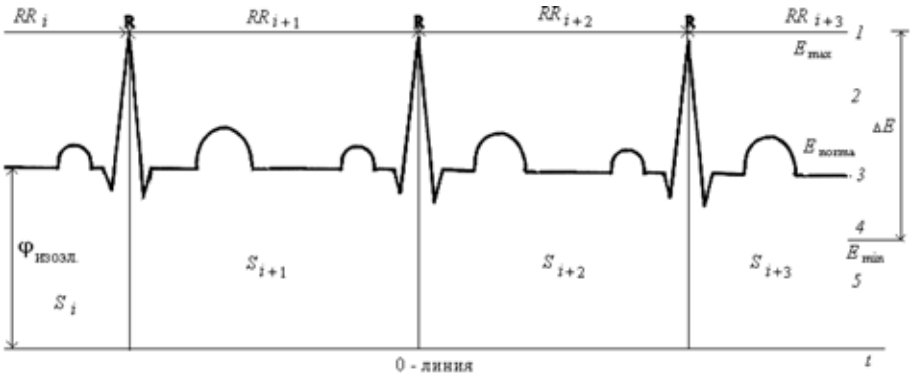

Рис. 1. Схема модельного представления соответствия эквипотенциального (E_{norma}) уровня изолинии и амплитуды (*R*) кардиоцикла:
1 — уменьшение частоты ритма; *2*, *4* — области изменения структуры кардиоцикла; *3* — средняя изоэлектрическая линия; *5* — увеличение частоты ритма; 0-линия — нулевая линия; ΔE — условный запас устойчивости структуры кардиоцикла

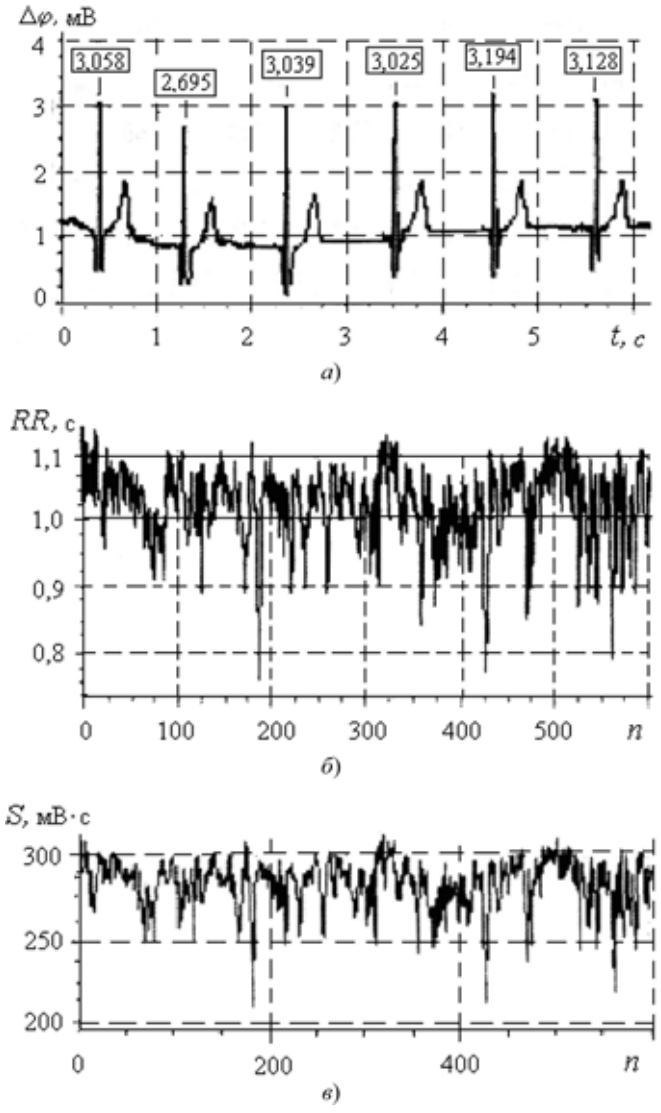

Рис. 2. Определение координат *R*-зубцов на фрагментарном участке ЭКГ (*а*) и построение *RR*- (*б*) и S_R-интервалограмм (*в*) по данным УЗО

разбивались на элементарные участки с длительностью времени дискретизации (Δx, мс). По технике расчета площадь S_{RR} под выделенным участком ЭКГ кардиоцикла равна сумме площадей прямоугольных трапеций высотой $\Delta x = 1$ мс. По смыслу это произведение среднеинтегрального потенциала кардиоцикла и соответствующего *RR*-интервала. Полученный цифровой ряд последовательности S_{RR}, названный S_R-интервалограммой (рис. 2, *в*), по технике получения является "оконной выборкой", а по смыслу является аналогом последовательности "фазовых энергетических ячеек" (ФЭЯ) на плоскости ЭКГ.

Программа определения координат пиков зубцов *R* на ЭКГ допускает ложные выделения или пропуски (до 1 %). Их можно исправить при обычном просмотре или синхронным делением, цифрового ряда S_R-интервалограммы на цифровой ряд *RR*-интервалограммы с формированием цифрового ряда значений S_R/RR. Таким образом, полную и системную информацию ЭКГ представляют пять цифровых рядов последовательностей текущих значений параметров ЭКГ: *RR*-интервалов, S_{RR}, *R*, а также относительных значений параметров S_R/RR и S_R/R, характеризующих относительный вклад зубцов *P* и *T* в значение площади кардиоцикла.

Сравнительная оценка соответствующих пар *RR*- и S_R-интервалограмм показала, что динамика интервалов времени с их топологической структурой почти совпадают (рис. 2, *б*, *в*). У групп УЗО значения коэффициентов корреляции *RR*- и S_R-интервалограмм находились в интервале значений 0,7...1. Высокое значение коэффициента корреляции (0,9...1) наблюдалось по записи ЭКГ в спокойном состоянии (положение "лежа"), а его уменьшение (0,7...0,89) в состоянии стресса (после экзамена). Для больных значение коэффициента корреляции, определенное по их *RR*- и S_R-интервалограммам, резко падает (ниже 0,7) [2, 8—10].

В норме значение *RR*-интервала оказывает более существенное влияние на значение S_{RR} по сравнению с амплитудным влиянием *QRS*-комплекса. Поэтому для здоровых людей динамика ФЭЯ практически не чувствительна к форме участков ЭКГ. При изменении уровня изолинии стремление сохранить связанную энергию морфологии комплексов приводит к адекватным изменениям амплитуды пиков, размеров интервалов и сегментов. Первым реагирует интервал общей паузы. Как оказалось — это своеобразный "люфт", обеспечивающий обратимость и устойчивость системного ритма. При ухудшении ФСО интервал общей паузы практически равен нулю, и искажения зубцов внутри кардиоцикла существенно и нелинейно меняют вклады их площадей в суммарную площадь. Поэтому S_R-интервалограмма становится информативной только при ухудшении ФСО.

Метод оценки вариабельности сердечного ритма

Информативным методом количественной оценки вегетативной регуляции сердечного ритма признан метод оценки вариабельности сердечного ритма (ВСР), как изменчивости *RR*-интервалов последовательных циклов сердечных сокращений при колебаниях тонуса вегетативной нервной системы. Анализ ВСР проводится у здоровых и больных людей во временной, частотной и частотно-временной параметрических областях [11]. Он включает определение и анализ показателей ВСР, каждый из которых отвечает за ту или иную сторону вегетативного влияния на ритм сердца.

Таблица 1

Временной анализ (уровень значимости $\alpha = 0,05$)

Величина	Норма. Запись 24 ч	Ритмограммы УЗО. Запись 20 мин		
		К, 29 дней	Ш, 34 дня	Группа — 32
SDNN, мс	141 ± 39	75,9 ± 11,4	50,1 ± 6,1	72,9 ± 8,9
AMo, %	34,3 ± 1,4	13,1 ± 1,6	16,3 ± 0,9	13,6 ± 2,4
ИН	53,9 ± 3,9	19,1 ± 4,1	33,6 ± 3,6	29,5 ± 12,9
RMSSD, мс	27 ± 12	41,8 ± 8,9	48,0 ± 8,0	43,3 ± 8,7
Triangular Index	37 ± 15	37,9 ± 6,6	28,7 ± 3,7	37,8 ± 4,4

Это позволяет использовать указанные показатели для оценки ФСО [11, 12].

При комплектовании групповой выборки значений по собственным выборочным данным возникает проблема назначения их объема *n*: либо по одинаковому интервалу времени регистрации ЭКГ, либо по одинаковому числу значений в цифровом ряду *RR*-интервалограммы. При заданной фиксированной длительности (20 мин) регистрации ЭКГ для группы здоровых молодых людей значения *n* могут отличаться более чем в 2 раза. Автором принята следующая позиция: если объектом исследования является последовательность интервалов времени, то объемы *n* определяются автоматически исходя из одинаковой длительности регистрации ЭКГ. Принятая позиция приводит к непреодолимым трудностям в применении статистического метода — основы метода ВСР, с точки зрения интерпретации расчетных данных для выборок разного объема, так как с изменением *n* значения параметров ВСР, вообще говоря, меняются (табл. 1).

Поэтому и с учетом тесных и устойчивых (вне зависимости от длин записей) параметрических корреляционных связей акцент в исследованиях диаграммы ритмов сердца (ДРС) был смещен в сторону группового анализа параметров. Параметр ВСР стандартного отклонения принят за основной управляющий параметр [11, 12], который при исследовании функциональных параметрических зависимостей, назначается аргументом — причиной изменения иных параметров ВСР (табл. 2 и см. рис. 4).

Временная область анализа ВСР. Для коротких двадцатиминутных записей в качестве зависимых параметров ВСР выбраны: амплитуда моды (AMo); индекс напряжения (ИН) регуляторных систем (стресс-индекс SI); квадратный корень из средней суммы квадратов разностей между соседними *RR*-интервалами (RMSSD) и общее число *RR*-интервалов, деленное на высоту гистограммы (триангулярный индекс). В табл. 1 приведены рекомендуемые нормальные значения [11, 12] и расчетные данные для экспериментальных групп УЗО.

Стандартное отклонение σ для первой групповой выборки (32 человека) молодых здоровых людей составляет 72,9 ± 8,9 мс, или раздельно: для юношей 71,5 ± 12,5 мс, для девушек 75,3 ± 12,0 мс.

При переходе от коротких записей к длинным значения основных показателей временной области возрастают в 2—3 раза с сохранением параметров формы и структуры полигона распределения ДРС (см. табл. 1). В табл. 2 приведены расчетные данные по назначенным к анализу параметрам и уравнения линий тренда (рис. 3) для первой группы УЗО.

Регистрации ЭКГ в первой группе проводились 2 раза в неделю в течение двух месяцев в интервале времени 5 ч. Тем не менее, все параметрические данные достоверно представляются функциональными зависимостями. Данные людей с "плохим" анамнезом смещены по функциональным кривым влево, а данные людей с "хорошим" анамнезом смещены вправо. Поэтому можно говорить о распределении групповых данных на функциональной кривой линии тренда. Вероятно, каждое уравнение (см. табл. 2) линии тренда (рис. 3, *а*, *б*, в) можно назвать "формулой ФСО" по соответствующим параметрам ВСР. Каждая ли-

Таблица 2

Функциональные связи параметров ВСР во временной области

Числовые и функциональные характеристики	Параметры ВСР (X)				
	ЧСС, уд/мин	RMSSD, мс	Triangular Index	AMo, %	ИН
Групповое значение $X \pm \Delta X$ Функция $X(\sigma)$ при достоверности аппроксимации	80,8 ± 4,5 $261\sigma^{-0,28}$ 0,58	43,3 ± 8,7 $7,7\mathbf{e}^{0,02\sigma}$ 0,86	37,8 ± 4,4 $28,8\ln(\sigma) - 83$ 0,88	13,6 ± 2,4 $724\sigma^{-1}$ 0,94	29,5 ± 12,9 $62\,800\sigma^{-1,9}$ 0,94
Групповое значение (юноши) $X \pm \Delta X$ Функция $X(\sigma)$ при достоверности аппроксимации	81,7 ± 6,5 $263\sigma^{-0,28}$ 0,59	42,0 ± 12,5 $7,1\mathbf{e}^{0,02\sigma}$ 0,85	37,4 ± 6,1 $28,5\ln(\sigma) - 82$ 0,92	14,4 ± 3,5 $681\sigma^{-1}$ 0,95	34,3 ± 19,8 $59\,310\sigma^{-1,9}$ 0,94
Групповое значение (девушки) $X \pm \Delta X$ Функция $X(\sigma)$ при достоверности аппроксимации	79,3 ± 5,4 $254\sigma^{-0,27}$ 0,50	45,5 ± 10,6 $10,2\mathbf{e}^{0,02\sigma}$ 0,93	38,6 ± 6,0 $29,2\ln(\sigma) - 88$ 0,80	12,3 ± 2,4 $785\sigma^{-1}$ 0,89	21,5 ± 9,2 $80\,740\sigma^{-2,0}$ 0,95

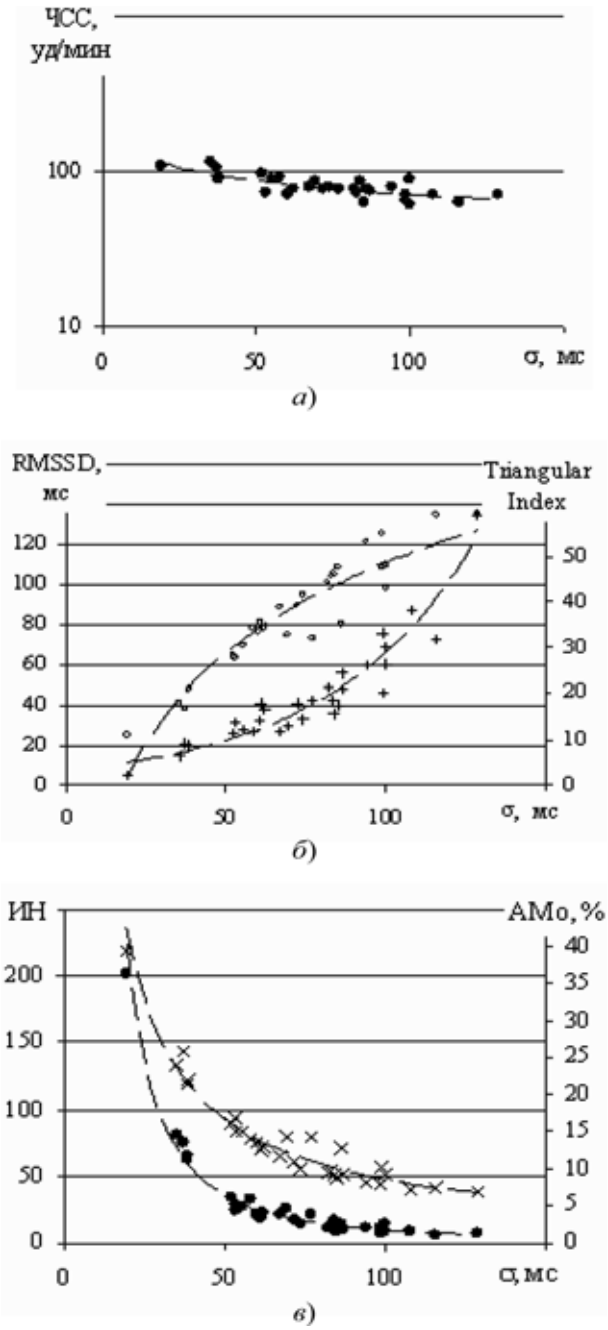

Рис. 3. Точечные графики: *а* — ЧСС(σ) (•); *б* — RMSSD(σ) (+), Triangular Index(σ) (○); *в* — ИН(σ) (•), AMo(σ) (×)

Таблица 3

Данные спектрального анализа (уровень значимости α = 0,05)

Величина	Норма. Запись 5 мин	Ритмограммы УЗО. Запись 20 мин		
		К, 29 дней	Ш, 34 дня	Группа — 32
TP, мс^2	3466 ± 1018	13 726 ± ± 3194	6196 ± 149	12 519 ± ± 2390
VLF, мс^2	1321 ± 399	5842 ± 1832	1424 ± 419	4202 ± 952
LF, мс^2	1170 ± 416	3214 ± 606	1153 ± 208	3377 ± 649
HF, мс^2	975 ± 203	1292 ± 467	778 ± 220	1684 ± 734
LF/HF	1,5—2,0	3,4 ± 0,9	2,3 ± 1,0	3,5 ± 0,8

ния тренда имеет признаки "шкалы ФСО" и интервала линейного распределения уровня ФСО. Увеличение объема выборки *n* приводит к смещению правой границы интервала линейного распределения без изменения формулы ФСО.

Частотная область анализа ВСР. В табл. 3 сведены данные норм стандартов (столбец 2) [12] и данные обработки ритмограмм для групп УЗО (столбец 3) по спектрам плотности мощности процедурой HRV программы "*EScreen*" [4]. Для короткой записи полная спектральная мощность (TP) равна сумме вкладов спектральных мощностей трех диапазонов частот: очень низких (VLF: 0,00333... 0,04 Гц), низких (LF: 0,04...0,15 Гц) и высоких (HF: 0,15...0,4 Гц) [12].

Размер области спектрального анализа определяется нижней и верхней граничными частотами. Если нижнюю частоту определяет длина записи, равная сумме *RR*-интервалов, представляющих *RR*-интервалограмму, то верхнюю частоту определяет *RR*-интервал кардиоцикла. Текущее значение *RR*-интервала переменно, а на горизонтальной оси *RR*-интервалограммы каждый из них обозначен номером отсчета. При спектральном анализе обычно принимают норму интервала кардиоцикла равной 0,8 с (75 уд/мин), получая 0,4 Гц. Становится очевидной условность назначаемой верхней границы общего частотного диапазона, определенных в единицах частоты (герцах). В указанном смысле условность нижней и верхней границ частотного диапазона исчезает, если их определять как $1/(2n)$. Тогда стандартная длина записи определяется не по шкале "стрелы времени", а по номерной шкале отсчетов — числом отсчетов *RR*-интервалов в их последовательности, т. е. объемом выборки *n*.

На графиках автокорреляционных функций ДРС УЗО и их спектров обнаружено, что все ритмограммы в той или иной степени содержат характерные колебания с периодами Δn = (10, 50, 100, 200, 300, 400, 600, 800, 1200). Колебания с малыми периодами Δn = (10, 50, 100) существуют цугами и могут следовать друг за другом с эффектом ассоциации, организуя участки пилообразной формы. Длина цугов растет с ростом периода, и наиболее "добротными" являются колебания с большими периодами Δn = (400, 600, 800, 1200) [2]. Таким образом, для разных УЗО, находящихся в интервале измерений в разных эмоциональном и физическом состояниях, обнаружены детерминированные сигналы в форме колебаний с *одинаковым* набором периодов. Если номерной ряд последовательности *R-R*-интервалов перевести во временной ряд, то шкала отсчетов станет неравномерной. Детерминированные сигналы в форме колебаний при этом сохранятся, но с *неодинаковым* набором периодов. Следовательно, ритм сердца имеет собственный внутренний ход времени с рит-

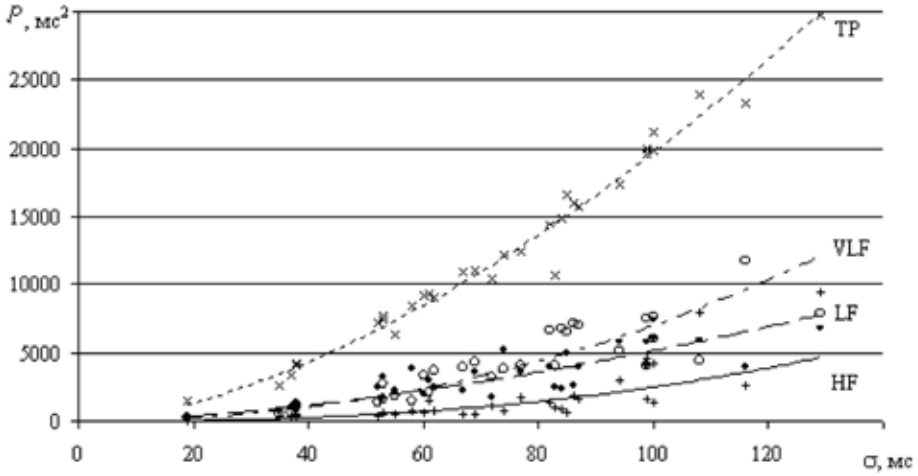

Рис. 4. Графики групповой зависимости четырех спектральных мощностей от σ. Показаны линии тренда: HF — (+), LF — (•), VLF — (○), TP — (×)

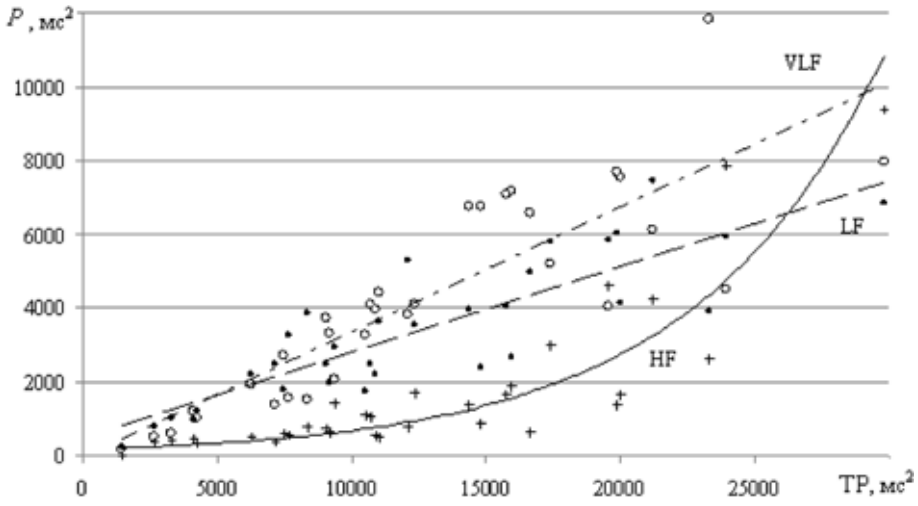

Рис. 5. Графики групповой зависимости трех спектральных мощностей по составляющим диапазонам частот, по степени их вклада в полную спектральную мощность (обозначения соответствуют рис. 4)

мической единицей измерения: *RR*-интервал — одна систола левого желудочка (один удар сердца). Существуют одинаковые ритмические интервалы, в течение которых происходят основные изменения в системе "сердце — регуляция".

К данным первой экспериментальной группы был применен тот же подход — групповой анализ параметров ВСР частотной области (рис. 4). Линии тренда показаны при максимальной достоверности аппроксимации ($R^2 > 0{,}8$). Соответствующие степенные функции имеют степень меньше 2.

На рис. 5 приведены точечные графики и соответствующие линии тренда зависимостей распределения полной спектральной мощности (TP) с учетом диапазона ультранизких (ULF: 0...0,00333 Гц) частот по вкладам трех составляющих спектральных мощностей: P_{VLF}(TP), P_{LF}(TP) и P_{HF}(TP).

При сравнении графиков на рис. 4 и 5 становится понятной причина отсутствия корреляционной связи между параметрами ВСР и параметром отношения $P_{\mathrm{LF}}/P_{\mathrm{HF}}$ [2, 11]. При очевидном сильном рассеянии данных (рис. 5) функция P_{HF}(TP) нелинейная в отличие от P_{LF}(TP).

Сравнивая данные табл. 3 для пяти- и двадцатиминутных записей, видно, что отношения P_{VLF} к другим составляющим P_{LF} и P_{HF} растут. Однако становится заметным (см. рис. 4), что значение TP значительно больше суммы трех составляющих спектральных мощностей: P_{VLF}(TP), P_{LF}(TP) и P_{HF}(TP). Очевидно появление превалирующего вклада мощности диапазона ULF. Увеличивая длину записи, а соответственно, и выборку, представляющую цифровой ряд ритма, в запись включаются новые все более низкие частоты процессов нелинейного влияния на ритм сердца. Увеличение амплитуд естественных процессов влияния с уменьшением их частоты может объяснить рост временных показателей с ростом длины записи ЭКГ.

Выводы

1. Согласно энергетической модели работы сердца каждому кардиоциклу ставятся в адекватное соответствие реализуемая в его собственном интервале времени потенциальная энергия и топологическая структура ЭКГ-комплексов. Стремление сохранить связанную энергию в морфологии комплексов приводит к адекватным изменениям амплитуды комплексов, размеров интервалов и сегментов.

2. Полную и системную ЭКГ информацию представляют пять цифровых рядов последовательностей текущих значений ЭКГ-параметров: *RR*-интервалов, S_{RR}, R, а также относительных значений параметров S_R/RR и S_R/R. При ухудшении ФСО искажения форм зубцов внутри *RR*-интервалов кардиоциклов существенно и нелинейно меняют их вклады в суммарную площадь и информативной становится S_R-интервалограмма.

3. Метод ВСР фиксирует уровень превалирования всех внесердечных влияний на ритм сердца над внутрисердечной регуляцией в форме откло-

нений значений параметров от условной нормы, а ВСР является фактором оценки отклика на внешние влияния.

4. Параметры ВСР ДРС группы УЗО или серии измерений одного человека вне зависимости от пола имеют функциональные зависимости, названные "формулами ФСО". Колебания размера *n* цифрового ряда *RR*-интервалов при рекомендуемой длине записи не влияют на формулу ФСО и слабо нелинейно влияют на значение интервала линейного распределения ФСО внутри группы (длину линии тренда) за счет смещений его правой границы. Вариабельность ритма сердца становится фактором ФСО при групповом анализе параметров ВСР.

5. Пространственно-временная комплектация ритмограмм по группе разных людей и по серии регистраций одного человека показала, с одной стороны, полную идентичность получаемых результатов (формул ФСО), а с другой стороны, что ВСР может быть фактором прогноза ФСО по данным временной серии регистрации ЭКГ.

Список литературы

1. **Физиология** человека. В 3 т. Т. 2.: Пер. с англ. под ред. Р. Шмидта и Г. Тевса. М.: Мир, 1996. 313 с.
2. **Кузнецов А. А.** Методы анализа и обработки электрокардиографических сигналов: Новые подходы к выделению информации. Владимир: Изд. ВлГУ. 2008. 140 с.
3. **Прилуцкий Д. А., Кузнецов А. А., Чепенко В. В.** Накопитель ЭКГ "AnnA Flash2000" // Методы и средства измерений физических величин. Н. Новгород: Изд-во НГТУ, 2006. С. 31.
4. **Medical** Computer Systems, Zelenograd, Moscow. URL: http://www.mks.ru.
5. **Кавасма Р., Кузнецов А., Сушкова Л.** Автоматизированный анализ и обработка электрокардиографических сигналов. Методы и система / Под ред. Л. Т. Сушковой. М.: Сайнс-пресс, 2006. 144 с.
6. **Мун Ф.** Хаотические колебания. Вводный курс для научных сотрудников и инженеров / Пер. с англ. Ю. А. Данилова и А. М. Шукурова. — М.: Мир. 1990. 312 с.
7. **Пригожин И.** От существующего к возникающему: Время и сложность в физических науках / Пер. с англ. под ред. Ю. Л. Климантовича. М.: Наука. 1985. 327 с.
8. **Кавасма Р. А., Кузнецов А. А., Сушкова Л. Т.** Новые методы обработки электрокардиографических сигналов // Биомедицинские технологии и радиоэлектроника. 2005. № 11—12. С. 12—20.
9. **Qawasma R. A., Sushkova L. T., Kuznetsov A. A.** Investigation of cardiovascular system with help of energy approach // Proc. of EMBEC'05, 3rd European Medical & Biological Engineering Conference: IFMBE European conference on Biological Engineering, Praque, Czech Republic, 2005.
10. **Кавасма Р. А., Кузнецов А. А., Сушкова Л. Т.** Энергетический и интерквантильный методы анализа электрокардиоинтервалов // Вестник новых медицинских технологий. 2005. Т. XII, № 3—4. С. 30—33.
11. **Баевский, Р. М., Берсенева А. П.** Введение в донозологическую диагностику. М.: Слово, 2008. 176 с.
12. **Heart** rate variability. Standards of measurement, physiological interpretation, and clinical use. Task Force of the Europian Society of Cardiology and the North American Society of Pacing and Electrophysiology // European Heart Journal. 1996. V. 17. P. 354—381.

УДК 004.9

В. П. Май, канд. техн. наук, вед. научн. сотр.,
e-mail: may@iacp.dvo.ru,
С. В. Мельман, ст. инженер-программист,
e-mail: gruzd@dvo.ru,
Институт автоматики и процессов управления
ДВО РАН, г. Владивосток

Система объемной визуализации объектов компьютерной томографии

Представлена система объемной визуализации объектов томографического исследования, позволяющая строить пространственные изображения костной или иной ткани, применительно к задачам челюстно-лицевой хирургии.

Ключевые слова: *компьютерная томография, визуализация объемов, методы сканирования, алгоритмы визуализации и обработки данных*

Введение

Современная медицинская диагностика в последние десятилетия пополнилась новым чрезвычайно мощным методом. Речь идет о компьютерной томографии [1—4], решающей задачи воссоздания послойного изображения внутренней структуры объекта исследования по совокупности проекционных данных, измеренных под многими ракурсами. Преимуществом этого метода, в отличие от других известных методов медицинской диагностики, является его весьма высокая информативность о каждом элементарном объеме исследуемого объекта. Наибольшее распространение в настоящее время получили рентгеновская компьютерная томография (РКТ) [5—7], в том числе спиральная компьютерная томография [8—10], и магнитно-резонансная томография (МРТ) [11—13].

Визуализация внутренних структур человека на основе данных томографического обследования получает все большее распространение в медицинской диагностике. Получаемые с помощью томографической аппаратуры четкие снимки множества сечений тела пациента позволяют с высокой достоверностью зрительно анализировать состояние различных органов и с привлечением воображения и опыта наблюдателя реконструировать их трехмерную структуру [14].

Альтернативой представления объектов поверхностями, как правило, полигональными, яв-

ляется представление объектов в виде объемов. Объем — это трехмерный массив кубических элементов (вокселей), представляющих единицы 3D-пространства. Объемная визуализация — это процесс получения на экране компьютера изображения, дающего наглядное представление о внутренней структуре объекта [15].

Если в обычной диагностической рентгенологии многие задачи диагноза могут быть достаточно точно решены при анализе двумерных изображений сечений, то в таких областях, как черепно-лицевая хирургия, имплантология, травматология, где терапевтическое решение по томографическим изображениям должен принимать не врач-рентгенолог, визуализация объема становится частью предоперационных процедур, помогая в планировании проведения хирургического вмешательства.

При планировании хирургических операций [16], в имплантологии [17], в черепно-лицевой хирургии [18] и в других случаях появляется необходимость рассмотрения объекта в целом со всеми его сложностями и дефектами. Возникает задача создания точных и реалистичных визуальных объемных представлений объектов по томографическим данным [19]. Например, при моделировании хирургической операции на виртуальном пациенте вначале он "оцифровывается" посредством томографических сканеров, затем строится и отрабатывается хирургическая операция на цифровой модели в виртуальной среде.

Исследования в этом направлении ведутся достаточно интенсивно, и с некоторыми из них можно ознакомиться, например, в работах [20—25]. Основная направленность этих исследований — расширение функциональности с обеспечением высокой степени наглядности структуры исследуемых объектов и создание дружественного интерфейса для пользователя. В качестве коммерческого продукта в последнее время приобрела известность система "3D-Doctor" американской компании *Able Software Corp* [26]. Эта программа имеет широкий спектр возможностей по трехмерному моделированию, обработке изображений, визуализации и анализу медицинских данных, в том числе результатов сканирования для РКТ и МРТ. Программа "3D-Doctor" поддерживает как черно-белые, так и цветные изображения, сохраненные в форматах DICOM, TIFF, Interfile, GIF, JPEG, PNG, BMP, PGM, RAW Image Data, а также создает поверхностные и послойные 3D-модели из двумерных послойных срезов в реальном времени. С помощью "3D-Doctor" можно проводить фильтрацию данных, очистку от шума, увеличивать резкость, контрастность (методами *image processing*), строить изоповерхности, сечения, проводить измерения, комбинировать *volume rendering* и визуализацию изоповерхностей.

Однако следует отметить, что программа "3D-Doctor", несмотря на свою привлекательность и универсальность, во-первых, является дорогостоящим коммерческим продуктом, во-вторых, от врача на ее освоение потребуется немало времени, в-третьих, ее использование может потребовать обновления технических средств.

В то же время разработка, о которой идет речь в данной статье, нацелена на выполнение узкого круга задач на имеющемся оборудовании. В связи с этим она проста в использовании, доступна и удовлетворяет требованиям специалиста челюстно-лицевой хирургии.

В данной статье предлагается один из подходов к разработке системы объемной визуализации объектов томографического исследования и созданию для врача-исследователя удобного интерфейса. Объемная визуализация дает возможность строить пространственные изображения костной или иной ткани для детального рассмотрения возможных ситуаций (переломов, опухолей) в различных ракурсах.

1. Получение данных о внутренней структуре объекта

Если РКТ практически является методом получения двумерного изображения, несмотря на отдельные попытки реконструкции трехмерного изображения, то для метода МРТ, напротив, трехмерность изображения заложена в самой его основе. Здесь можно по выбору регистрировать данные от всего трехмерного объекта либо только от одного его среза. Вместе с тем, РКТ и МРТ используют совершенно одинаковые принципы автоматического, управляемого компьютером сканирования, обработки и получения послойного изображения внутренней структуры органов.

Данные сканирования приводятся, в конечном итоге, к пространственной регулярной 3D-решетке скалярных данных. Существует несколько следующих традиционных методов визуализации.

- *Метод изоповерхности*, при котором осуществляется визуализация специально построенной изоповерхности как геометрического места точек, где заданная в объеме полученных данных функция равна некоторому выбранному, так называемому пороговому, значению. Недостатком метода является то, что видна лишь часть объема, принадлежащая поверхности, но зато она хорошо воспринимается врачом для детального и подробного рассмотрения.
- *Метод прямого объемного рендеринга* основан на непосредственном отображении всего объема

данных, разделенного на элементарные частицы (точки), проецируемые на экран монитора.

Недостатком метода является "захламление" изображения многочисленными частицами, проецируемыми на один и тот же пиксель на экране. Преимущество же метода состоит в том, что управляя прозрачностью и цветом точек, можно получать картину изображения высокого качества.

- *Метод трассировки лучей* заключается в пропускании луча через каждый пиксель на экране. При этом окраска пикселя вычисляется по лучу, проходящему сквозь объем. Преимуществом этого метода перед другими является возможность наиболее точного отображения объема на плоскость экрана, однако возникает сложность в правильной интерпретации значений при переводе в цвет и прозрачность на пересечении луча и объема.

Все методы объединяет один подход — визуализация только тех вокселей, которые пользователь выделяет как значимые. В медицинских приложениях основными объектами визуализации человеческого организма выступают внутренние органы, кости, кожа, которые неплохо выделяются с помощью метода, основанного на создании изоповерхностей.

Для процесса сканирования целесообразно выбрать формат DICOM — международный формат данных, используемый в медицинской практике. В этом формате может храниться любая информация, касающаяся пациента, лечащего врача, истории болезни, рецептов, данных УЗИ и рентгена, вплоть до измерений температуры тела во время болезни. Этот формат имеет собственные сетевые протоколы для работы с удаленным сервером данных. Медицинские томографы результаты сканирования записывают также в формате DICOM, используя при этом малую часть всего формата, т. е. записывается только время сканирования, ФИО (врача, пациента, медсестры). Сама структура формата DICOM древовидная, что позволяет объединять несколько последовательных сканов в одно "дерево" для пациента. Таким образом, если бы на одном сервере хранились данные по пациенту в течение всей его жизни, то и история болезни пациента, вместе с диагнозом и комментариями врача, могла бы храниться на электронном носителе как единое целое. Так, процесс сканирования в нескольких словах, без технических подробностей, можно описать следующим образом: аппарат получает "скан плотности" тканей человека в плоскости, перпендикулярной оси позвоночника. При этом "кушетка", на которой лежит пациент, непрерывно движется. Этот скан получается в виде растра (например, 512 × 512 пикселей), каждый пиксель сканируется в течение какого-то времени, и если подходить к скану строго, то все пиксели скана фактически имеют небольшой сдвиг, поэтому принято считать, что у одного скана есть толщина (за время одного скана кушетка продвигается на 2 мм, значит и толщина скана 2 мм). Далее весь процесс сканирования записывается в виде набора плоских сканов, где каждый пиксель — это целое значение от 0 до 2000. В DICOM-структуре фиксируется время сканирования, разрешение скана, толщина скана, число сканов, и набор параметров, позволяющих рассчитать физические размеры каждого пикселя, при этом сохраняется пространственная привязка к объекту сканирования.

2. Отображение данных в трехмерном пространстве

Скалярные данные в пространстве можно представить как $s: M \times I \to R$, где M — множество данных; I — временной промежуток, в котором представлено данное множество. Если говорить о реальном трехмерном статическом, не меняющемся во времени, наборе данных, то это есть преобразование пространственной координаты $X(x, y, z)$ с помощью функции в скаляр $F(X) \to s$, $(s \in R)$.

Трехмерная модель объекта, как такового, не строится, т. е. выполняется визуализация изоповерхности, одной или нескольких, на наборе пространственных данных, но идентификация объектов не проводится.

Имея набор сканов и параметры привязки к физическим параметрам объекта, можно каждый пиксель на скане интерпретировать как воксель в пространстве.

Пусть заданы параметры для сканирования головы пациента, размеры (ш. в. д.: 20 × 30 × 25 см), сканирование с разрешением скана 512 × 512 пикселей и толщина скана 2 мм, тогда физические размеры вокселя: 200 мм/512 = 0,39; 300 мм/512 = = 0,586; 0,39 × 0,586 × 2,0 мм. Число сканов при этом будет: 250 мм/2 мм = 125 шт.

Таким образом, результатом сканирования объекта является пространственная решетка размерности 512 × 512 × 125 с физическими размерами 20 × 30 × 25 см.

3. Алгоритмы компьютерной визуализации и обработки данных

Алгоритм построения изоповерхности. Для конвертирования воксельного представления изоповерхностей в полигональное был реализован алгоритм "марширующих кубиков", что обеспечило аппаратную поддержку визуализации изоповерхностей скалярных полей. Этот алгоритм предназначен для генерации полигонального приближения изоповерхности некоторой скалярной функ-

ции $F(x, y, z)$, заданной на равномерной 3D-сетке значений [27]. Получаемая поверхность является аппроксимацией изоповерхности некоторого уровня L, т. е. геометрического места точек, удовлетворяющих условию $F(x, y, z) = L$.

Для упрощения алгоритма строится функция

$$F^0(x, y, z) = F(x, y, z) - L.$$

Таким образом, задача сводится к построению изоповерхности F^0 на уровне 0:

$$F^0(x, y, z) = 0.$$

Алгоритм работает следующим образом. Пусть имеется регулярная 3D-решетка размерности $[i, j, k]$, тогда вершины V_{ijk} этой решетки образуют $(i-1)\cdot(j-1)\cdot(k-1)$ ячеек (кубиков). Таким образом, кубик может либо не содержать часть изоповерхности, что выполняется при условии, когда все вершины кубика одного знака (ноль считать положительным), либо содержать часть поверхности. Тогда:

- перебирая все ячейки, ставим им в соответствие байт состояния, где каждый бит указывает на знак вершины ячейки: 0 — положительная вершина (значение $F^0(x, y, z) = 0$ тоже следует считать положительным), 1 — отрицательная вершина;
- ячейки с байтом состояния 0 и 255 пустые, а для остальных по заранее составленным таблицам отыскивается набор треугольников, которые аппроксимируют изоповерхность, проходящую внутри ячейки;
- координаты точек пересечения искомой изоповерхности и ребер ячеек вычисляются линейной интерполяцией.

Табличный выбор треугольников, аппроксимирующих изоповерхность, дает достаточно хороший результат при большой скорости построения изоповерхности в целом. Применительно к визуализации данных медицинского томографа алгоритм маршрирующих кубиков удовлетворяет всем требованиям: быстрое построение изоповерхности, достаточно высокое качество.

Алгоритм обработки данных. Данные медицинского томографа хранятся в виде 3D-массива целых чисел от 0 до 2000. Наблюдая изменение параметров порогового значения на пространственных скалярных данных медицинского томографа, исследователю (медработнику) важно видеть отдельные ткани (например костные) и построение изоповерхности с пороговым значением, равным эквиваленту костной ткани, что позволяет выделить эту самую ткань для исследования (в случае перелома). При обычном разрешении медицинского томографа 512 × 512 пикселей на срез и 125 срезах (~ 32,8 млн вокселей) описанный алгоритм маршрирующих кубиков позволяет построить изоповерхность (на обычном компьютере AMD Athlon 1800XP+) за ~ 0,1...0,5 с (разброс характеризуется сложностью изоповерхности).

Так как врач работает с данными медицинского сканирования, то резонно предположить, что материалом для исследования будет человеческое тело, для которого основные пороговые параметры можно свести в специальную таблицу, в которой наиболее важные значения параметров будут сопровождаться информацией об изоповерхности, которая будет построена при выборе соответствующего параметра (например, кость, кожный покров, зубы).

4. Программный продукт для построения трехмерной модели объекта

Исходя из требований, предъявляемых врачами к визуализации данных сканирования медицинского томографа, был разработан программный продукт узкой специализации, предназначенный для визуализации костной ткани для планирования операций при переломах челюстно-лицевого скелета. Особенностью данных операций является высокая сложность диагностики трещин и травм. Посредством медицинского томографа проводится сканирование поврежденного участка, затем с помощью специального программного обеспечения осуществляется реконструкция и визуализация поврежденного костного скелета для дальнейшего анализа хирургом.

Требования к программному продукту:

- работа с форматом данных DICOM медицинского томографа;
- удобный интерфейс для работы с древовидной структурой DICOM;
- просмотр сканов в режиме реального времени;
- построение изоповерхности по пороговому значению, заданному в интерактивном режиме медицинским работником для тонкой настройки визуализируемой костной поверхности;
- свободное вращение объекта;
- возможность смещения источника света.

5. Возможности разработанной системы

С учетом выдвинутых требований был спроектирован программный продукт "TomoView3D" на MSVC++. Графическая часть реализована с использованием библиотек OpenGL. С помощью данной системы осуществляются следующие операции.

1. *Работа с древовидной структурой формата DICOM*, которая ведется с помощью бесплатной библиотеки DCMTK 3.5.3. В начале работы пользователь открывает директорию с данными о пациенте, программный продукт (ПП) предлагает древовидную структуру со всеми томографиче-

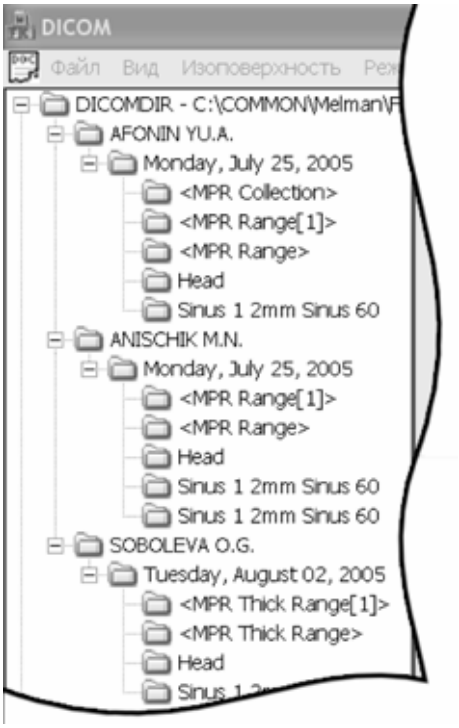

Рис. 1. Данные о пациенте

скими исследованиями, с указанием даты и области сканирования (рис. 1). Пользователь может предварительно просмотреть сканы (рис. 2, см. вторую сторону обложки), и если выбор состоялся, то можно приступать к исследованию объемной реконструкции изоповерхностей.

2. *Построение изоповерхности* — выполняется в отдельном элементе интерфейса, где находится "ползунок", перемещение которого определяет выбор порогового значения. Рядом отображаются минимальное, максимальное и текущее значения. При перемещении ползунка программный продукт в реальном времени перестраивает пространственную реконструкцию изоповерхности, и на экране (в четырех областях) демонстрируется новая изоповерхность (в четырех ракурсах) (рис. 3, см. вторую сторону обложки).

3. *Визуализация срезов на наборе данных*, при которой четвертая область демонстрирует свободный срез (плоский) исследуемого объекта любой плоскостью (рис. 4, см. вторую сторону обложки).

4. *Свободное вращение объекта*, которое происходит в каждой из четырех областей визуализации. Изменение ракурса (навигация) осуществляется с помощью мыши (левая кнопка = вращение объекта вокруг центральной точки, колесико мыши = приближение и удаление).

5. *Интерактивное смещение источников освещения*, оно происходит при зажатой правой кнопке мыши (от исходного положения у наблюдателя), что позволяет рассматривать мелкие детали объекта.

6. *Многооконный интерфейс*, предоставляющий дополнительные удобства пользователю для работы одновременно с несколькими наборами данных в разных окнах.

7. *Сохранение вида модели для вывода на печать*, когда интерфейс позволяет сохранить любую из четырех областей вида объекта (или все четыре одновременно) в графический файл BMP с последующим выводом на печать.

Заключение

Программный продукт "TomoView3D" позволяет работать с полученными от медицинского томографа данными в их исходном виде и предоставляет интерфейс работы с данными, схожий с интерфейсом программного обеспечения самого томографа. Это облегчает освоение специалистами нового инструмента визуализации томографических данных при вводе его в эксплуатацию. "TomoView3D" дает возможность в режиме реального времени выбирать на множестве скалярных данных медицинского томографа пороговые значения при определении наиболее удобного и оптимального для врача-исследователя варианта объемной визуализации.

Программный продукт "TomoView3D" использовался в Дальневосточном окружном медицинском центре Росздрава в практике челюстно-лицевой хирургии для анализа характера и сложности повреждений при переломах в предоперационный период и в процессе последующего наблюдения.

Работа выполнена при финансовой поддержке ДВО РАН в рамках Программы П2 фундаментальных исследований Президиума РАН, проект 09-I-П2-05.

Список литературы

1. **Физика** визуализации изображений в медицине. В 2-х томах. Т. 1 / Пер. с англ. под ред. С. Уэбба. М.: Мир, 1991. 408 с.
2. **Календер В.** Компьютерная томография. М.: Техносфера, 2006. 344 с.
3. **Терновой С. К., Сапожкова Л. П.** Методы лучевой диагностики. М.: Феникс, 2007. 138 с.
4. **Хофер М**. Компьютерная томография. Базовое руководство. 2-е изд. М.: Медицинская литература, 2008. 244 с.
5. **Верещагин Н. В., Брагина Л. Б., Вавилов С. Б., Левина Г. Я.** Компьютерная томография мозга. М.: Медицина, 1986. 256 с.
6. **Коновалов А. Н., Корниенко В. Н.** Компьютерная томография в нейрохирургической клинике. М.: Медицина, 1988. 346 с.
7. **Помозгов А. И., Терновой С. К., Бабий Я. С., Лепихин Н. М.** Томография грудной клетки. Киев: Здоровье, 1992. 288 с.
8. **Рабухина Н. А., Голубева Г. И., Перфильев С. А.** Спиральная компьютерная томография при заболеваниях челюстно-лицевой области. М.: МЕДпресс-информ, 2006. 128 с.
9. **Морозов С. П., Насникова И. Ю., Синицын В. Е.** Мультиспиральная компьютерная томография. М.: ГЭОТАР-Медиа, 2009. 132 с.
10. **Прокоп М., Галански М.** Спиральная и многослойная компьютерная томография. В 2-х томах. Т. 2. М.: МедПресс, 2009. 712 с.
11. **Коновалов А. Н., Корниенко В. Н., Пронин И. Н.** Магнитно-резонансная томография в нейрохирургии. М.: Видар, 1997. 560 с.

12. **Ринк П. А.** Магнитный резонанс в медицине. М.: Геотар-Мед, 2003.
13. **Синицын В. Е., Устюжанин Д. В.** Магнитно-резонансная томография. М.: ГЭОТАР-Медиа, 2008. 208 с.
14. **Мошнегуц С. В., Барбараш Л. С., Журавлева И. Ю.** Трехмерная визуализация как средство эффективного анализа данных низкопольной МРТ // Вестник рентгенологии и радиологии. 2005. № 3. С. 43—46.
15. **Ягель Р.** Рендеринг объемов в реальном времени // Открытые системы. 1996. № 5 (19). С. 28—33.
16. **Федоров В. Д., Кармазановский Г. Г., Гузеева Е. Б., Цвиркун В. В.** Виртуальное хирургическое моделирование на основе данных компьютерной томографии. М.: Видар, 2004. 184 с.
17. **Жигун В. В.** Компьютерная томография. 2007. URL: http://www.implant.ru/content/view/186/367/
18. **Стоматологическая** клиника. ООО "Клиника Партнер". 2008. URL: http://www.clinicpartner.ru/library/implantaciyazubov/mestaja_diagostika_pri_z uboji/trexmernajavizualizatsija_lit.html.
19. **Поммерт А., Пфлессер Б., Риемер М., Шиеманн Т., Шуберт Р., Тиеде В., Хон К. Х.** Визуализация объема в медицине // Открытые системы. 1996. № 5.
20. **Blackwell M., Nikou C., DiGioia A. M., Kanade T.** An Image Overlay system for medical visualization // Transactions on Medical Image Analysis. 2000. V. 4. P. 67—72.
21. **Dohi T.** Surgical Robotics and Three Dimensional Display for Computer Aided Surgery // Proc. of Computer Aided Radiology and Surgery, CARS 2000, San Francisco, U. S. A, June 2000. P. 715—719.
22. **Jansen T., Rymon-Lipinski B., Krol Z., Ritter L., Keeve E.** An Extendable Application Framework for Medical Visualization and Surgical Planning // Proc. of SPIE Medical Imaging, MI'01, San Diego, CA. 2001. February 17—23.
23. **Zachow S., Lamecker H., Elsholtz B., Stiller M.** Reconstruction of mandibular dysplasia using a statistical 3D shape model // Proc. Computer Assisted Radiology and Surgery. 2005. P. 1238—1243.
24. **Subburaj K., Suresh Kumar P., Bansal D., Ravi B.** Virtual Orthopaedic Surgery System // International Conference on Total Engineering, Analysis and Manufacturing Technologies, Bangalore. 2007. Oct. 4—6.
25. **Chunbo Bao, Boliang Wang.** A Open Source Based General Framework for Virtual Surgery Simulation // International Conference on BioMedical Engineering and Informatics. 2008. BMEI. V. 1. P. 575—579.
26. **Able** Software Corp. URL: http://www.3d-doctor.com
27. **Lorensen W. E., Cline H. E.** Marching cubes: A high resolution 3D surface construction algorithm // Computer Graphics (SIGGRAPH '87 Proceedings). 1987. V. 21. P. 163—169.

УДК 004.94

А. В. Петрухин, канд. техн. наук, доц.,
А. В. Золотарев, аспирант,
Волгоградский государственный
технический университет,
e-mail: olorint@gmail.com

Методика автоматизации начальных этапов процесса проектирования биомеханических систем

Современное развитие информационных технологий обеспечивает возможность применения методов автоматизации проектирования технических систем в новых областях, в частности, при проектировании биомеханических систем. Проектирование биомеханических систем является сложным процессом, требующим анализа биологических тканей, выбора функционального назначения системы, ее кинематической схемы, а также определения узлов и деталей, из которых она состоит. В работе рассматривается методика автоматизации начальных этапов проектирования биомеханических систем. Полученные результаты позволяют повысить качество проектирования биомеханических систем за счет автоматизации начальных этапов проектирования.

Ключевые слова: *автоматизация, проектирование, биомеханические системы*

Исследования, проводимые в области биомеханики, охватывают различные уровни организации живой материи: ткани, органы, системы органов и др. Описание механических явлений, происходящих в тканях и системах человека, представляет большой интерес при проведении исследований в области биомеханики. В настоящее время биомеханика развивается по нескольким направлениям, среди которых, помимо спортивной биомеханики, можно выделить: инженерную биомеханику, связанную с роботостроением; медицинскую биомеханику, исследующую причины, последствия и способы профилактики травматизма, прочность опорно-двигательного аппарата, вопросы протезостроения; эргономическую биомеханику, изучающую взаимодействие человека с окружающими предметами в целях их оптимизации. Различные области биомеханики исследуют поведение специализированных биомеханических систем [1].

Биомеханической системой (БМС) будем называть систему взаимодействующих биологических и технических объектов. Время взаимодействия элементов между собой определяет один из характеристических признаков БМС. На основе этого признака можно выделить два класса БМС: постоянные и временные. Как правило, основной целью использования постоянных БМС является замещение полностью утраченных функций биологических составляющих системы. Ярким примером класса постоянных БМС являются различного вида экзо- и эндопротезы. Применение постоянных БМС предполагает отсутствие возможности восстановления утраченных функций биологических элементов. В отличие от постоянных БМС, временные БМС по-

зволяют полностью или частично заменить функции биологических элементов. К временным БМС относятся зубные скобы, лангеты, чрескостные аппараты. Постоянное увеличение сферы применения биомеханических систем приводит к необходимости решения проблемы совершенствования методов их проектирования, которая имеет важное социально-экономическое значение и обеспечивает сокращение сроков и улучшение исходов лечения. Рассмотрим основные этапы процесса проектирования биомеханических систем на примере проектирования чрескостных аппаратов (ЧА).

Проектирование ЧА выполняется как на уровне разработки концепции нового изделия, так и на уровне детального проектирования. На начальном этапе проектирования необходимо провести структурный синтез технического объекта (ТО) и определить основные параметры и характеристики создаваемого объекта. Также на данном этапе проектирования необходимо определить конструктивно-функциональную схему (КФС) создаваемого объекта. Наиболее часто для построения КФС используется принцип двухуровневой иерархии, рассмотренный в работе [2]. КФС используется для разработки различных видов реализаций ЧА. Число технических элементов, соответствующих различным элементам выбранной схемы, равно нескольким сотням единиц.

Функциональные возможности конкретной реализации ЧА для выбранной схемы определяются составом элементов, используемых в работе. Выбор реализации, наиболее полно соответствующей поставленным задачам, достаточно сложен. Анализ методов решения технических изобретательских задач, приводимых в работах [2—5], показывает, что для решения данной задачи наиболее оптимальным является применение метода морфологического анализа и синтеза технических решений (МАиСТР). Для каждого элемента КФС составляется список альтернативных вариантов реализации конструктивных элементов. Анализ вариантов реализации ЧА, проводимый с использованием метода МАиСТР, позволяет получить новые конструктивные реализации ЧА и отдельных его элементов.

Изменение физического принципа действия ТО также позволяет получать новые виды реализации ЧА. Для реализации этой возможности необходимо построить потоково-функциональную схему (ПФС) системы в соответствии с алгоритмом, описанным в работах [2—5]. Для проектирования новых видов ЧА используется метод синтеза ФПД системы на основе заданной физической операции. На данном этапе целесообразно использовать словарь технических функций. После этого выполняется синтез цепочки физико-технических эффектов (ФТЭ), используемых для реализации ФПД системы. Наращивание цепочки проводится с учетом качественной оценки совместимости ФТЭ на основе анализа входов и выходов элементов. Цепочки, полученные в результате описанных действий, подвергаются анализу количественной совместимости элементов. Полученные варианты ФПД используются для формирования различных видов принципиальных схем аппаратов. Для проектирования принципиальной схемы аппарата могут быть применены и другие методы проектирования технических систем, описанные в работах [2—8].

Полученные принципиальные схемы аппарата используют для проектирования и сборки индивидуализированных реализаций ЧА. На данном этапе проектирования ЧА выполняется ряд базовых операций, определяющих целевое назначение подэтапов: разработка индивидуализированной математической модели, описывающей состояние биологических тканей; расчет базовых параметров процесса лечения, проводимого с использованием биомеханической системы; расчет геометрических параметров; компоновка индивидуализированной реализации системы. Совокупность параметров, описывающих пространственное положение участка ткани, ее геометрическая форма, а также совокупность физических параметров биологических тканей определяют состояние биологических тканей.

Для выявления базовых свойств и законов, описывающих поведение ЧА в условиях его взаимодействия с внешней средой, необходимо провести биомеханический анализ ЧА. Для проведения биомеханического анализа необходимо проанализировать и описать текущее состояние тканей, методы анализа которых могут различаться в зависимости от целей. Результаты диагностики состояния тканей лежат в основе математической модели, описывающей состояние выбранного участка тканей. В соответствии с работой [9] математическая модель — это "эквивалент" объекта, отражающий в математической форме важнейшие его свойства — законы, которым он подчиняется, связи, присущие составляющим его частям, и т. д.". Под индивидуализированной математической моделью участков тела пациента будем понимать совокупность данных о состоянии тканей участка тела пациента, взаимосвязи между различными участками тканей и законы, описывающие процесс деформации тканей.

Разработка индивидуализированной модели участков тела пациента позволяет моделировать реальные процессы, происходящие при оперативном лечении, без перехода к усредненным моделям и процессам. Для того чтобы разработанная

математическая модель позволяла проектировать индивидуализированные реализации ЧА, необходимо предусмотреть возможность моделирования состояния тканей в процессе лечения на основе данной модели. Математическая модель также должна содержать не только базовую информацию, хранящуюся в результатах диагностики, но и поддерживать возможность хранения дополнительной информации, описывающей, например, совокупность физических характеристик биологических элементов. Математическая модель, разработанная в соответствии с этими требованиями, позволит хранить информацию об индивидуальных параметрах пациента, моделировать процесс лечения и проектировать индивидуализированную реализацию ЧА.

На основе информации, содержащейся в индивидуализированной математической модели, выполняется расчет основных параметров процесса лечения. Для расчета параметров процесса лечения необходимо описать предполагаемое состояние математической модели до лечения и после. Под состоянием индивидуализированной математической модели будем понимать совокупность значений параметров, определяющих состояние тканей участка тела пациента в определенный момент времени.

Состояние модели "до лечения" описывается информацией, полученной в результате предварительной диагностики пациента. Для получения состояния модели "после лечения" необходимо изменить состояние модели таким образом, чтобы оно соответствовало состоянию тканей после устранения деформации. В связи с тем, что в результате применения чрескостных аппаратов, как правило, производится изменение геометрических характеристик тканей, моделирования процесса деформации тканей аналогично моделированию процесса деформации геометрической формы трехмерного объекта, которое не представляет особой сложности. На данном этапе форма и компоновка аппарата не оказывают влияние на параметры процесса лечения. Основные параметры процесса лечения определяются в рамках сферы применения ЧА. При вычислении основных параметров операции учитываются физико-анатомические особенности строения тканей. Несмотря на то, что параметры процесса лечения, проводимого с использованием биомеханической системы, например, в ортодонтии, будут существенно отличаться от параметров процесса лечения, проводимого в ортопедии, список параметров процесса лечения не меняется в пределах одной области применения биомеханических систем. На основе значений параметров процесса лечения выполняется расчет индивидуализированной реализации системы. Основной задачей этого этапа является не только расчет геометрических параметров чрескостных аппаратов, но и выбор элементов, на основе которых осуществляется компоновка технических элементов. На первоначальном этапе выбирают схему аппарата, которая определяет функциональное назначение аппарата и позволяет провести его предварительную компоновку. В процессе предварительной компоновки определяют геометрические параметры главных и второстепенных элементов аппарата. На основе геометрических параметров главных элементов, а также базы типоразмеров элементов проводят выбор реализаций элементов, на основе которых выполняется сборка аппарата. Определение геометрических параметров и методов соединения главных элементов аппарата позволяет ввести ограничения на применение его второстепенных функциональных узлов. Используя информацию, содержащуюся в математической модели, можно визуально представить состояние тканей в области крепления второстепенных узлов, что позволяет выбрать наилучшие их положения. После определения положения второстепенных функциональных узлов проводится конечная компоновка аппарата, результатом которой является его кинематическая схема, список деталей определенных типоразмеров, используемых при создании аппарата. Таким образом, создается индивидуализированная реализация чрескостного аппарата.

Разработанную методику применяют для проектирования индивидуализированных аппаратов чрескостного остеосинтеза, в частности, ее использовали сотрудники Волгоградского государственного медицинского и Волгоградского государственного технического университетов при проведении совместных научно-практических работ. Приведенная методика автоматизации начальных этапов проектирования биомеханических систем позволяет выполнять проектирование индивидуализированных реализаций различных видов биомеханических систем. Определение требований, предъявляемых к математической модели, используемой для представления информации о состоянии биологических тканей, позволяет не только повысить точность расчета параметров биомеханической системы, но и обеспечить визуальное сопровождение процесса проектирования биомеханической системы. Повышение точности расчета параметров процесса лечения дает возможность увеличить эффективность применяемой биомеханической системы. Автоматизация процесса компоновки биомеханической системы позволяет избежать возможных ошибок при создании конечных реализаций биомеханических систем.

Список литературы

1. **Бернштейн Н. А.** Общая биомеханика (Основы учения о движениях человека). М.: Изд. РИО ВЦСПС, 1996.
2. **Половинкин А. И.** Основы инженерного творчества. М.: Машиностроение, 1988. 368 с.
3. **Автоматизация** поискового конструирования (искусственный интеллект в машинном проектировании). / Под. ред. Половинкина А. И. — М.: Радио и связь, 1981. 344 с.
4. **Половинкин А. И.** Теория проектирования новой техники: закономерности техники и их применение (монография). — М.: Информэлектро, 1991.
5. **Половинкин А. И.** Методы инженерного творчества. — Волгоград, 1984. — 365 с.
6. **Фоменков С. А., Петрухин А. В., Давыдов Д. А.** Автоматизированная система концептуального проектирования технических объектов и технологий // Информатика — машиностроение. 1999. № 3. — С. 7—10.
7. **Голдовский Б. И., Вайнерман М. И.** Комплексный метод поиска решений технических проблем. — М.: Речной транспорт, 1990.
8. **Ulrich K. T., Eppinger S. D**. Product Design and Development. Irwin McGraw-Hill, 2000.
9. **Самарский А. А., Михайлов А. П.** Математическое моделирование. Идеи. Методы. Примеры. — 2-е изд., испр.. — М.: Физматлит, 2001. — 320 с.

Поздравляем юбиляра!

Главному научному сотруднику Института проблем информатики РАН, академику РАЕН, действительному члену Международной академии наук высшей школы, доктору технических наук, профессору

Константину Константиновичу КОЛИНУ,

известному ученому, крупному специалисту в области современных информационных технологий, члену редколлегии ряда отечественных научных журналов, включая журнал "Информационные технологии", исполняется 75 лет.

Окончив в 1959 году Ленинградскую военно-воздушную академию им. А. Ф. Можайского, он прошел огромный жизненный и творческий путь от инженера по радиотехнике до научного руководителя ведущего академического института в области информационных технологий и вычислительных систем.

Талантливый ученый и преподаватель Константин Константинович известен как один из создателей и инициаторов развития в России нового научного направления информатики — социальной информатики, а также как видный организатор, внесший неоценимый вклад в развитие современной национальной доктрины России в сфере образования и информатизации. Результаты исследований К. К. Колина изложены в 12 книгах и учебных пособиях и более 250 статьях. За цикл работ в области социальной информатики награжден Президиумом РАЕН серебряной медалью им. акад. П. Л. Капицы.

Высокий профессионализм, широкая эрудиция, большая трудоспособность, чуткость, отзывчивость и любовь к Родине снискали ему большое уважение учеников и коллег.

Дорогой Константин Константинович!
Поздравляем Вас с юбилеем и желаем Вам крепкого здоровья, большого счастья, новых творческих успехов в Вашей многогранной деятельности!

Редколлегия и редакция журнала.

8—12 марта 2010 года в г. Дрезден (Германия) с успехом прошла тринадцатая международная конференция DATE 2010 (Design Automation and Test in Europe) — крупнейшая в Европе комплексная конференция и выставка, объединившая академических исследователей, разработчиков микроэлектронных систем и устройств, продавцов, заказчиков и пользователей систем автоматизированного проектирования микроэлектронных схем.

В этом году программа конференции состояла из 83 секций, включая 17 специализированных, и была посвящена рассмотрению всех вопросов, связанных с проектированием и технологическими решениями встраиваемых микро- и наноэлектронных систем, включая маршруты проектирования, архитектурные решения, тестирование готовых комплексных систем, разработку программных средств и примеры выполненных проектов встраиваемых систем с высокой степенью интеграции.

Вместе с конференцией прошла выставка, на которой были представлены оборудование и программные продукты для разработки, проектирования и тестирования встраиваемых систем и систем-на-кристалле, заказных ИС, программируемых вентильных матриц и печатных плат. Особое внимание было уделено представлению успешных проектов и разработок использования наноэлектронных решений в автомобильной и авиационной промышленностях.

На форум 2010 года было прислано 984 доклада (рис. 1). Рост числа присланных докладов наглядно показывает интерес к этой конференции международного научно-технического сообщества в текущих кризисных экономических условиях, сложившихся на мировом и Европейском рынках. Из них были отобраны 344 лучших, что составляет около 35% от числа представленных. Это свидетельствует о высочайшем рейтинге конференции и ее качестве. В конференции приняли участие специалисты из 39 стран мира.

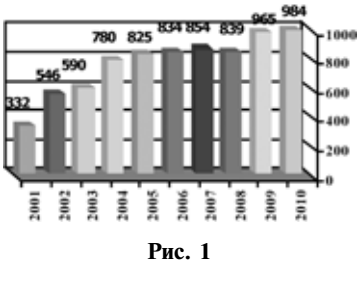

Рис. 1

Рис. 2

Рис. 3

Конференция открылась 8 марта. 11 учебно-практических курсов, прошедших в этот день, были посвящены вопросам проектирования и производства энергоэффективных телекоммуникационных систем и прикладным аспектам использования наноэлектронных систем в авиа- и автомобилестроении.

Основная программа конференции началась 9 марта с церемонии открытия конференции Генеральным председателем Технического комитета DATE 2010, проф. Джованни де Микелли (Giovanni De Micheli), Политехнический университет г. Лозанна, Швейцария. На первой сессии конференции был представлен обзорный доклад проф. Санжованни-Вичентелли (Alberto Sangiovanni-Vincentelli), профессора Калифорнийского университета, г. Беркли и члена Совета директоров компании Cadence (рис. 2), посвященный вопросам и опыту совместных международных разработок сложных микроэлектронных систем, выполняемых виртуально, т. е. без личного ежедневного контакта разработчиков, а также проблемам менеджмента подобных проектов. Вторым докладом пленарной сессии был доклад г-на Германа Юл (Herman Eul), Вице-президента компании Infineon, посвященный истории, текущим тенденциям и будущему беспроводных средств связи (рис. 3).

С 9 по 11 марта прошли основные секции конференции (рис. 4), по широчайшему кругу вопросов проектирования ИС, системной интеграции микроэлектронных схем, систем и устройств, разработки средств САПР, технических решений для нанометровых проектов, тестированию проектных решений и т. д.

В последний день конференции были проведены 5 семинаров, посвященных проблемам разработки архитектурных и технологических решений для перенастраиваемых систем, проектированию встраиваемых параллелизованных многоядерных вычислительных платформ, проблемам тестирования схем на кремнии, результатам кооперации участников 7-ой Рамочной программы Европарламента в сфере информационных технологий и т. д.

Большинство докладов конференции были представлены участниками из США и стран Западной Европы. В 2010 г. огромный рост числа присланных докладов продемонстрировал Китай.

В выставке приняли участие более пятидесяти компаний: как ведущих, так и ряд небольших компаний и университетских групп, предлагающих уникальные новые продукты и разработки. Выставка DATE пользуется большой популярностью. Число ее фирм-экспонентов и посетителей растет год от года. В этом году общее число посетителей конференции и выставки превысило 2700.

В рамках выставки прошли презентации новых программных продуктов и системных решений ведущих мировых продавцов САПР.

Рис. 4

Растущее число участников конференции и выставки наглядно подтверждает тот факт, что DATE является не просто обычной европейской конференцией по проблемам САПР и проектирования ИС, а действительно глобальным научным событием мирового масштаба.

Приглашаю ученых и разработчиков микроэлектронной аппаратуры России к участию в следующей конференции и выставке, которая должна состояться 14—18 марта 2011г. в г. Гренобль (Франция). Информация о ней содержится на сайте www.date-conference.com

По всем вопросам обращайтесь по адресу:
ИППМ РАН, 124681, Москва, ул. Советская, д. 3; тел./факс. (499) 729-9208

А. Л. Стемпковский, член комитета спонсоров DATE,
директор ИППМ РАН, академик РАН

VII Международная научно-практическая конференция
ИННОВАЦИИ НА ОСНОВЕ ИНФОРМАЦИОННЫХ И КОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Качество - Безопасность - Диагностика

Инфо 2010

Сочи

с 1 по 10 октября

НАПРАВЛЕНИЯ РАБОТЫ

- Инновационные информационные и коммуникационные технологии в образовании
- Научно-исследовательские инновационные проекты
- Инновационные информационные и коммуникационные технологии в технике и промышленности
- Инновации в экономике и социальной сфере

ЗАЯВКА НА УЧАСТИЕ В КОНФЕРЕНЦИИ

Заявку на участие в конференции можно оформить на сайте www.diag.ru до 15 июня 2010г.

СООРГАНИЗАТОРЫ

- МЕЖДУНАРОДНАЯ АКАДЕМИЯ ИНФОРМАТИЗАЦИИ
- ИНСТИТУТ ПРОБЛЕМ УПРАВЛЕНИЯ РАН
- ГНИИ ИТТ «ИНФОРМИКА»
- ИНСТИТУТ ИНФОРМАТИЗАЦИИ ОБРАЗОВАНИЯ РАО
- ЕВРОПЕЙСКИЙ ЦЕНТР ПО КАЧЕСТВУ
- ФГУП «МКБ «ЭЛЕКТРОН»
- МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ
- МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ЭКОНОМИКИ, СТАТИСТИКИ И ИНФОРМАТИКИ
- МОСКОВСКИЙ ИНСТИТУТ РАДИОТЕХНИКИ, ЭЛЕКТРОНИКИ И АВТОМАТИКИ
- МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПРИБОРОСТРОЕНИЯ И ИНФОРМАТИКИ
- МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ЛЕСА
- МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ
- СОЧИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТУРИЗМА И КУРОРТНОГО ДЕЛА
- ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
- СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
- СУРГУТСКИЙ ИНСТИТУТ МИРОВОЙ ЭКОНОМИКИ И БИЗНЕСА «ПЛАНЕТА»
- СТУДЕНЧЕСКИЙ ИННОВАЦИОННО-НАУЧНЫЙ ЦЕНТР

СЕКРЕТАРИАТ ПРОГРАММНОГО И ОРГАНИЗАЦИОННОГО КОМИТЕТОВ

По вопросам участия в конференции обращаться:
109028, г. Москва, Б. Трехсвятительский пер., д.3, МИЭМ, каф. РТУиС.
Телефоны: 8(926)-383-07-40, 8(903)-782-08-65, 8(916)-905-91-43, 8(926)-744-10-14.
E-mail: conf@diag.ru Сайт конференции: http://diag.ru
Увайсов Сайгид Увайсович, Иванов Илья Александрович

CONTENTS

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 *Т. В. Зверева*

Сдано в набор 10.03.2010. Подписано в печать 20.04.2010. Формат 60×88 1/8. Бумага офсетная. Печать офсетная. Усл. печ. л. 9,8. Уч.-изд. л. 11,80. Заказ 343. 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