ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

1(149)
2009

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

УДК 004.3

А. Б. Барский, д-р техн. наук, проф.,
МИИТ
E-mail: arkbarsk@mail.ru

Применение логических нейронных сетей для выбора оптимальной стратегии обслуживания потока запросов в системе GRID-вычислений

Предлагается адаптивная система динамического принятия решений при выборе оптимальной стратегии обслуживания потока запросов на выполнение вычислительных работ Центром GRID-технологий. Критерием оптимизации является максимальная загрузка вычислительных средств, обеспечивающая максимальную пропускную способность Центра, а также соблюдение директивных сроков обслуживания. Назначение вычислительных средств, соблюдение приоритетов и синхронизация выполнения взаимозависимых запросов осуществляются с помощью логических нейронных сетей на основе текущих и пролонгированных характеристик потока запросов и состояния средств системы.

Ключевые слова: *система принятия решений, поток запросов, оптимальная стратегия обслуживания, логическая нейронная сеть, GRID-вычисления.*

Сними-ка, Елдырин, с меня пальто...
Ужас, как жарко! Должно полагать, перед дождём...
.....
Надень-ка, брат Елдырин, на меня пальто...
Что-то ветром подуло... Знобит...
А. П. Чехов. Хамелеон

Введение. Адаптивная система обслуживания запросов на основе динамического выбора оптимальной стратегии управления

Рассмотрим обобщенную адаптивную схему динамического оптимизированного распределения потока запросов между каналами в многоканальной системе массового обслуживания (рис. 1, см. третью сторону обложки). Критерием оптимизации является достижение максимальной загрузки каналов, служащей максимальной пропускной способностью системы обслуживания [1].

В основе диспетчера, распределяющего запросы-задания, лежат несколько решающих правил, каждое из которых реализует некоторую стратегию обслуживания. Решающие правила различаются по скорости реализации и по достигаемому эффекту в части статистических оценок обеспечения высокой пропускной способности. Выбор решающего правила и локализация его применения во времени зависят от характеристик потока запросов: от переменной плотности, от их типового состава, приоритета, частичной упорядоченности.

В этом случае задача диспетчирования становится трудно формализуемой. Оперативное управление текущим выбором стратегии необходимо выполнять с помощью логической нейронной сети, на рецепторный слой которой подаются текущие и пролонгированные характеристики потока запросов, а также текущие параметры состояния системы обслуживания. Нейроны выходного слоя максимальным возбуждением указывают на решающее правило, которым следует пользоваться в дальнейшем. То есть, максимально возбужденный нейрон выходного слоя инициирует запуск соответствующей процедуры распределения в составе диспетчера.

Адаптация обеспечивается возможностью перезакрепления решающих правил за нейронами и вводом в рассмотрение новых правил — по статистическим оценкам эффективности распределения (распараллеливания).

Выделение неизменного состава характеристик, участвующих в принятии каждого решения, служит основой построения простейших по структуре совершенных логических нейронных сетей, где передаточная функция является аналогом конъюнкции и осуществляет суммирование значений входных сигналов. Поскольку такая нейронная сеть является однослойной, то на деле реализуется простейшая схема "голосования" в пользу той стратегии, которая в наибольшей степени адекватна сложившейся ситуации — текущим характеристикам потока запросов и состоянию системы обслуживания.

Рассмотренная схема легко развивается, модифицируется и специализируется.

Модель обслуживания запросов Центром GRID-технологий

Общий принцип адаптивного выбора оптимальной стратегии распределения запросов в многоканальной системе обслуживания требует детального рассмотрения для конкретных предложений. Определяющую роль при решении задач распределения играют не столько физическая природа каналов обслуживания, сколько характе-

ристики потока запросов и потока обслуживания. Не зависящими от этой природы можно считать и критерии оптимизации — максимальную загрузку каналов и минимизацию времени или соблюдение директивных сроков обслуживания.

Поток запросов отражает "физический смысл" задачи на абстрактном уровне и диктует конкретное содержание тех *решающих правил*, что лежат в основе диспетчера распределения. Поэтому конкретизация задачи неизбежна.

Рассмотрим ее на примере гипотетического Центра GRID-технологий (рис. 2, см. четвертую сторону обложки), выполняющего работы вычислительного характера на основе потока запросов, поступающих из WWW.

Будем считать, что основой вычислительных средств Центра является ЛВС — локальная вычислительная сеть, процессоры (рабочие станции — РС) которой являются каналами обслуживания. Однако следует отметить существенное расширение исследуемой проблемы: для выполнения некоторых запросов, инициирующих решение задач высокой сложности, используется не одна РС, а несколько — *сегмент* ЛВС для параллельного решения задачи. Это расширяет представление об единичных каналах обслуживания, допускает их комплексирование.

Более того, динамическое сегментирование ЛВС проводится для текущего закрепления за сегментами отдельных типов запросов. Текущие характеристики потоков запросов различных типов с учетом тенденций их изменения и их совместного анализа требуют пролонгированного закрепления ресурсов для обслуживания этих типов запросов. Такое закрепление должно учитывать приоритет типа, временные ограничения, текущую занятость средств, сервис.

Можно ограничить задачу, считая, что выделение сегментов ЛВС проводится не динамически, а жестко связано с решением некоторых задач высокой сложности, выполняемых методом конечных разностей, линейного и целочисленного линейного программирования, динамического программирования, транспортной задачи и др.

Однако в произвольном случае Центр может участвовать в договорных обязательствах в составе систем автоматического (или автоматизированного) управления. Это также предъявляет требования к сегментации его ЛВС, к выделению и ограничению доступа к сегментам, к обеспечению высшего приоритета и режима секретности. Такие потоки запросов к сегментам ЛВС выделяются из общего потока запросов.

Таким образом, общий поток запросов, поступающий из WWW, формируется на основе четырех потоков и концентрирует всю полноту возможных типов запросов, специфику и количество требуемых средств, частичную упорядоченность заданий, их приоритеты, временные оценки и ограничения, тенденции изменения (дрейф) характеристик потоков заданий и вычислительных средств.

Все указанные характеристики (возможно, и другие) должны быть систематизированы, классифицированы, градуированы — чтобы образовать полное и непротиворечивое факторное пространство событий, достаточное для динамического, в реальном времени, принятия решений об оптимальной стратегии обслуживания потока запросов.

Нейросетевое управление ресурсами

Диспетчер ресурсов в каждом i-м такте работы системы совместно анализирует характеристики четырех потоков ($i = 1, \ldots, 4$) на основе их относительного приоритета, используя следующие динамически формируемые данные:

- усредненное по серии последних запросов время $\Delta t_i^{(j)}$ между запросами j-го типа;
- аналог первой производной $d_i^{(j)} = \Delta t_i^{(j)} - \Delta t_{i-1}^{(j)}$, способствующей пролонгации тенденции;
- текущий приоритет поступившего запроса по отношению к поступившим запросам других потоков;
- объем запрашиваемых ресурсов;
- наличие необходимых ресурсов с учетом перспективы их освобождения;
- директивный срок выполнения запроса.

Эти значения, с учетом достоверности их принадлежности частным интервалам, подаются на рецепторный слой логической нейронной сети. По комбинациям значений возбуждения рецепторов возбуждаются нейроны выходного слоя, указывающие на варианты назначения ресурсных сегментов ЛВС, процессоров суперкомпьютера или отдельных рабочих станций для выполнения запросов. При этом определяется и стратегия ожидания освобождения необходимых ресурсов в случае их занятости. Принимаемые решения, в частности, могут быть следующих видов:

- выделить m процессоров;
- выделить n процессоров с предоставлением инструментария;
- выделить p процессоров для параллельного решения задачи из состава пакета прикладных программ (ППП);
- назначить очередную рабочую станцию;
- начать накопление необходимого ресурса по мере его освобождения;
- продолжить накопление ресурса для задержанного запроса и т. д.

Следует предположить, что логическая нейронная сеть, разрабатываемая для управления ресурсами, будет совершенной. Ведь при каждом обращении к ней учитываются все перечисленные факторы, и она является однослойной.

Обучение нейронной сети первоначально происходит на основе рекомендаций экспертов, а впоследствии — на основе опыта эксплуатации Центра.

Очевидно, что запросы потока 1 (рис. 2, см. четвертую сторону обложки) всегда обладают самым высоким приоритетом. Для ликвидации большой задержки приоритеты потоков 2, 3 и 4 могут быть плавающими.

Нейросетевое обоснование стратегии распараллеливания

Распараллеливание вычислений [1] выполняется в том случае, когда производительности одиночных средств — компьютеров, PC, процессоров суперкомпьютера — недостаточно для решения задачи.

Организация динамического распараллеливания вычислений — диспетчирование — является основным трудоемким и ответственным элементом построения управляемого вычислительного процесса в многопроцессорных вычислительных системах и вычислительных комплексах на основе сегментов ЛВС.

Основная трудность диспетчирования заключается в частичной упорядоченности (во времени выполнения) работ, которая присуща системам управления в реальном времени.

Диспетчер распараллеливания входит в состав операционной системы компьютера и сам по себе увеличивает "накладные расходы" производительности на собственную реализацию во времени. Поэтому быстродействие диспетчера является важнейшим требованием к нему.

Другим требованием, предъявляемым к диспетчеру распараллеливания, является достаточная близость формируемых им планов к точным оптимальным, полученным на основе решения задачи распараллеливания в адекватной постановке.

Дело в том, что решаемая диспетчером задача относится к важному классу задач исследования операций [2] — к классу задач параллельного программирования. Это задачи экспоненциальной сложности (NP-сложные задачи). Решение таких задач для множеств частично упорядоченных работ весьма трудоемко и никак не может быть положено в основу программ операционной системы компьютера. Поэтому прибегают к эвристическим методам полиномиальной сложности (P-сложность) для оперативного, динамического распараллеливания. В основе таких методов можно выделить одно или более *решающих правил*, погруженных в алгоритм распараллеливания и обусловленных предпочтительным выбором альтернативы в ключевых ситуациях.

Применение каждого решающего правила при распараллеливании влечет различный объем вычислений — накладных расходов. Однако, как правило, увеличение сложности алгоритмов распараллеливания обусловлено попытками приблизить результаты планирования к точным оптимальным. Поэтому по результатам достаточного опыта можно считать, что более трудоемкий диспетчер статистически обеспечивает планы параллельного выполнения работ, более близкие к оптимальным.

При практическом решении задач распараллеливания ограничиваются однородными системами выделяемых средств (однородными вычислительными системами [1]), а различные подходы к диспетчированию различаются вариантами единственного решающего правила.

По трудоемкости, а следовательно, "по оптимальности", варианты решающего правила можно упорядочить следующим образом.

1. Из множества работ, выполнение которых может начаться в текущий момент времени, назначение на свободные процессоры следует осуществлять произвольно (по сложившемуся порядку номеров, первую в списке, случайно и т. д.).

2. Из множества работ, выполнение которых может начаться в текущий момент времени, назначение на свободные процессоры следует осуществлять в порядке невозрастания времени выполнения работ (в первую очередь назначать работу с максимальной оценкой времени выполнения).

3. Из множества работ, выполнение которых может начаться в текущий момент времени, в первую очередь на свободный компьютер (процессор) назначать работу, предшествующую максимальному объему непосредственно или транзитивно следующих ей работ, включая ее саму.

4. При известном директивном сроке выполнения комплекса работ (в том числе — в системе реального времени) из множества работ, выполнение которых может начаться в текущий момент времени, назначать на свободный компьютер (процессор) работу, обладающую минимальным значением позднего срока окончания выполнения.

5. В дополнение к правилу 4: При равных минимальных значениях позднего срока окончания выполнения в первую очередь назначить работу в соответствии с правилом 3.

В различных случаях применения эффективно используются и другие комбинации решающих правил 2, 3, 4.

Первое и второе решающие правила используются в динамических диспетчерах, предполагающих выполнение назначенных работ одновременно с дальнейшим планированием.

Третье и последующие решающие правила предполагают работу диспетчера вне времени выполнения назначенных работ: диспетчер разрабатывает план выполнения этих работ, затем этот план реализуется без включения диспетчера. Это обеспечивает плотную загрузку диспетчеров при

выполнении работ и, следовательно, их скорейшее освобождение — переход в свободный ресурс.

Очевидно, что чем большим ресурсом времени обладает диспетчер в связи с "разреженным" потоком высокоприоритетных заданий при заданных директивных сроках выполнения, тем больше искушение использовать решающие правила для распределения этих заданий, приближающие результаты планирования к оптимальным. Ведь при этом достигается минимальное время выполнения совокупности взаимозависимых работ, плотно загружаются процессоры, выделяется время для фонового выполнения запросов с меньшим приоритетом. Это справедливо, если одновременное увеличение накладных расходов на планирование в совокупности с разработанным планом все же служит оптимизации параллельного вычислительного процесса, т. е. минимизации времени выполнения комплекса работ высокого приоритета.

Вместе с тем, очевидно, что при плотном потоке заданий невысокого приоритета требование их оперативного распределения между вычислительными средствами заслоняет собой все другие требования к оптимизации вычислительного процесса. Становится не до оптимизации: ведь плотная загрузка всех средств и так гарантирована! В этом случае достаточно использовать простейшее решающее правило.

Таким образом, только на основе моделирования и опытных исследований можно для достаточного числа точек факторного пространства — характеристик потоков заданий, их временных характеристик, директивных сроков их выполнения, требуемых ресурсов и состояния загрузки вычислительных средств — получить ряд рекомендаций по применению решающих правил в диспетчере распараллеливания.

Так формируется основа логической нейронной сети, примерный вид которой приведен на рис. 3 (см. третью сторону обложки). Связи устанавливаются на основе моделирования или опытных данных. Предполагается, что обслуживается высокоприоритетный поток задач оперативного планирования и управления (например, в системе реального времени), для которого резервирование вычислительных средств проводится заранее с помощью диспетчера ресурсов.

Директивный срок T определяется как длина отрезка времени, через которое распределяемое множество частично упорядоченных работ должно быть выполнено обязательно. Ресурс директивного срока при планировании распараллеливания определяется как разность:

$$\Delta T = T - \frac{1}{n} \sum_{j=1}^{m} t_j,$$

где n — число выделенных процессоров; m — число распределяемых работ.

На рецепторном слое указывается принадлежность характеристик интервалам значений или значениям (объем используемых ресурсов). Максимальное возбуждение нейрона выходного слоя (при высоком пороге) указывает на необходимость автоматического включения соответствующего решающего правила в диспетчер.

Формирование гипотез для обучения нейронной сети

Возможность оптимизации совместной реализации запросов на выполнение нескольких частично упорядоченных работ предполагает частичное или полное накопление этих запросов (работ) в некотором "окне просмотра" диспетчера. Диспетчер с высокой частотой анализирует содержимое "окна просмотра", пытаясь "уложить" выполнение этих работ за минимальное время, не превышающее директивный срок T. Время выполнения запросов, а также время работы диспетчера измеряются условными единицами. Временные оценки известны заранее.

Пусть для выполнения работ выделены два процессора суперкомпьютера. Диспетчер реализован на управляющем процессоре — на HOST-процессоре.

В "окне просмотра" накопились работы, частичную упорядоченность (порядок следования) которых можно описать информационным графом (рис. 4).

Пусть диспетчер использует решающее правило 1. Время его однократного выполнения составляет 1 у. е., $T = 13$ у. е.

Тогда временная диаграмма выполнения комплекса работ может быть представлена на рис. 5.

Для диспетчера, использующего решающее правило 2, временная диаграмма выполнения того же комплекса работ на двух процессорах представлена на рис. 6.

Временная диаграмма выполнения того же комплекса взаимосвязанных работ для диспетчера, реализующего решающее правило 4, при заданном директивном сроке окончания выполнения $T = 13$ у. е. представлена на рис. 7. Учтено, что время однократной работы диспетчера для двух выделенных процессоров составляет 3 у. е.

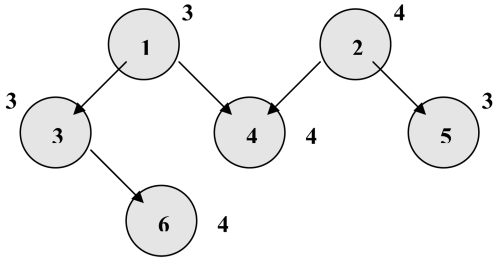


Рис. 4. Информационный граф

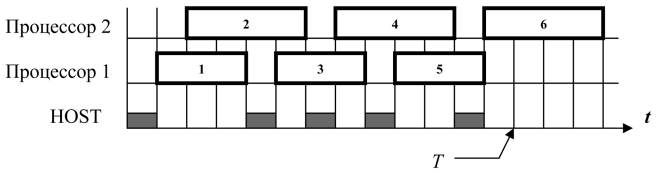


Рис. 5. Временная диаграмма выполнения работ при решающем правиле 1

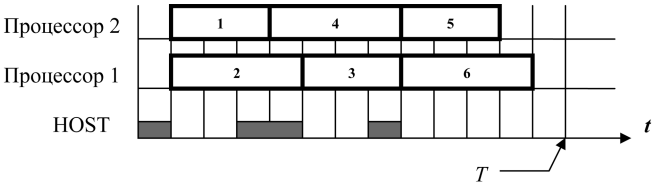


Рис. 6. Временная диаграмма выполнения работ при решающем правиле 2

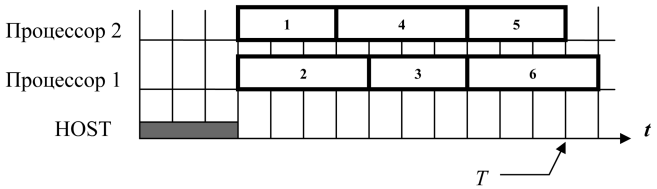


Рис. 7. Временная диаграмма выполнения работ при решающем правиле 4

На основе результатов применения решающих правил 1, 2 и 4 в диспетчере можно сформулировать гипотезу экспертной рекомендации для записи связей (трассировки) в логической нейронной сети, представленной на рис. 3 (см. третью сторону обложки). Для этого необходимо найти среднее время t выполнения работы, интенсивность потока запросов на отрезке $[0, T]$ и ресурс директивного срока ΔT. Интенсивность потока в примере принимается постоянной.

Такая гипотеза может быть следующей:

для $t \subset [3, 4]$, $\Delta T \subset [2, 4]$, $\Delta t_i^{(\Gamma)} \subset [0{,}4, 0{,}5]$, $d_i^{(\Gamma)} \subset [0, 0{,}001]$ и для двух процессоров следует считать предпочтительным применение в диспетчере решающего правила 2.

Заключение

Почему предлагаемая система названа адаптивной?

Во-первых, потому, что в *рабочем* режиме она, принимая решение, учитывает, приспосабливается к складывающейся обстановке на основе принципов *дедуктивного* мышления.

Во-вторых, существует возможность развития этой системы в режиме *обучения*, возможность все большего применения принципов *индуктивного* мышления на основе накапливаемого опыта функционирования. Конечно, на первом этапе разработки и эксплуатации анализ этого опыта, влияющего на повышение эффективности выбора решений и на сам арсенал этих решений, принадлежит разработчику-"учителю". Однако возможности дальнейшей автоматизации этого процесса неисчерпаемы.

Список литературы

1. **Барский А. Б.** Параллельные информационные технологии: Учебн. пос. М.: ИНТУИТ; БИНОМ. Лаборатория знаний, 2007. 503 с.

2. **Барский А. Б.** Нейронные сети: распознавание, управление, принятие решений. М.: Финансы и статистика, 2004. 175 с.

3. **Барский А. Б.** Логические нейронные сети: методика построения и некоторые применения // Информационные технологии. Приложение. 2006. № 8. 32 с.

4. **Барский А. Б.** Параллельные информационные технологии в основе Grid-системы // Информационные технологии. 2006. № 12. С. 54—60.

УДК 004.057.4

В. И. Шкунов, аспирант,
Нижегородский государственный технический университет

О методологии оценки и сравнения характеристик различных протоколов беспроводных сетей с переменной топологией

Рассматривается методология сравнения различных характеристик беспроводных сетей. Формулируется проблема общего подхода к таким вопросам, приводятся некоторые результаты.

Ключевые слова: *сеть, моделирование, беспроводной, методология.*

Введение

Одной из обычных задач, которые решаются в ходе какого-либо исследования, как правило, является задача качественной и количественной оценки существующих решений в исследуемой области.

Выработка единой методологии и соответствующих инструментов/способов оценки существующих и будущих решений является необходимым этапом в разработке какого-либо решения. Такая методология позволяет объективно сравнивать существующие аналоги, а также проводить

оценку разрабатываемых прототипов на самых ранних стадиях, что позволяет устранять недочеты тут же, в самом процессе разработки.

Беспроводные сети с переменной топологией (в англоязычной литературе — *ad-hoc* сети) не являются исключением.

Задача

Выбрать из существующих/адаптировать/разработать новую методологию сравнения и оценки таких характеристик *ad-hoc* сетей как: PDF (*Packet Delivery Ration*) — отношение числа принятых пользовательских пакетов к переданным; NRL (*Normalized Routing Load*) — отношение числа переданных служебных пакетов к числу принятых пользовательских, среднее время доставки пользовательского пакета, пропускной способности и других.

Тестовые платформы

Сформулированная выше задача тесно связана с вопросом разработки и применения тестовых платформ и отдельных инструментов, позволяющих обеспечить объективную оценку тех или иных решений [1, с. 15]. Взаимосвязь проявляется в том, что результаты, получаемые в результате таких тестов/испытаний, могут быть использованы для расчета широкого спектра характеристик моделируемых сетей. С этой точки зрения поставленную задачу можно переформулировать (свести) к следующей задаче: выбрать из существующих/адаптировать/разработать новую тестовую платформу/отдельный инструмент, определить условия проведения тестов и разработать определенный набор тестов. Результаты, полученные на выходе, будучи обработаны согласно определенным процедурам, позволят рассчитать экспериментальные характеристики моделируемых сетей.

Принято все тестовые платформы/инструменты разделять на симуляторы, эмуляторы и действующие прототипы сетевых устройств. Основные отличия заключаются в том, на какой стадии разработки решения они используются.

Симуляторы позволяют посредством имитационного моделирования получить определенные результаты на самом раннем этапе разработок, когда можно реализовывать определенные алгоритмы и сценарии работы сети на одном из популярных языков программирования.

Эмуляторы подразумевают тестирование программного кода, написанного непосредственно для действующего прототипа сетевого устройства, т. е. на определенном узкоспециальном языке программирования с учетом всех соответствующих ограничений и который может быть в последствии без изменений использован в реальных устройствах.

Действующие прототипы сетевых устройств. Тесты с их использованием проводятся на завершающей стадии разработки решения, когда первые две ступени — применение стимуляторов и эмуляторов — успешно пройдены. Очевидно, что этот вид тестовых платформ самый дорогой и сложный в использовании. Обзор одной из действующих сетей, состоящих из прототипов сетевых устройств, описан в работе [1, с. 25].

У каждого типа тестовых инструментов есть свои преимущества и недостатки. В данном случае, когда речь идет о модификации/разработке нового протокола очевидно необходимо начать с проверки полученного решения с помощью симуляторов.

Универсальная тестовая платформа

В настоящий момент положение вещей таково, что, как правило, каждый из существующих симуляторов решает специфический набор задач, для которого и был разработан. Очевидно, что такой подход как разработка отдельного инструмента для отдельной задачи бывает часто целесообразнее, чем попытка приспособить к решению этой задачи уже существующий инструмент, который не подходит по каким-то причинам. Вместе с тем, когда возникает задача, поставленная выше, т. е. необходимость сравнения результатов, полученных для однотипных задач разными инструментами, приходится признавать, что все они, как правило, использовали отличающуюся методологию.

Сравнительный анализ симуляторов

Наиболее популярные симуляторы на сегодняшний день: ns2 (*The University of California, Berkeley*), GloMoSim (*The University of California, Los Angeles*), OPNET Modeler (*Massachusetts Institute of Technology*), VANS (*Osaka University*), APE, JiST/SWANS, J-Sim, OMNeT++, QualNet, SNS.

Выбирая из представленных выше симуляторов по таким основным критериям, как область применения, поддерживаемые протоколы и принцип моделирования, можно сделать вывод, что некоторые из них, например, SNS, являются модификациями более универсальных пакетов (в случае SNS — ns2) и предназначены для работы в некоторых специфичных условиях: при больших диаметрах сети и большом числе узлов сети (до нескольких тысяч). Другие не имеют такого количества поддерживаемых протоколов уровня PHY/MAC как ns2: IEEE 802.11a/b/g, UWB [2]. Ну и, наконец, ns2 поддерживается большим числом энтузиастов, постоянно выпускается в новых версиях и является свободно распространяемым си-

мулятором. Принимая во внимание все вышесказанное, в качестве единой тестовой платформы можно выбрать симулятор ns2.

Условия проведения тестов

Помимо единой тестовой платформы необходимо определиться с набором тестов, применимых ко всему множеству испытуемых сетей, характером трафика, способами анализа полученных результатов.

Модель распространения радиоволн. На данный момент существует несколько таких моделей, основные из которых следующие:

- shadow model;
- two-way ground reflection model;
- free space reflection model.

Наиболее распространены две последние.

Модель перемещения узлов (mobility model). Эта модель задается следующими параметрами: [-nn nodes] [-p pausetime] [-s maxspeed] [-t simtime] [-x maxx] [-y maxy]. В порядке перечисления: число узлов; пауза между движениями узла; максимальная скорость перемещения узла; время моделирования; координаты прямоугольника, в пределах которого могут перемещаться узлы сети. Симулятор ns2 позволяет описывать перемещения узлов мобильной сети в трехмерной системе координат: *X*, *Y*, *Z*. На практике используется двумерная (плоская) система координат такая, что $Z = 0$. Такая модель задания закона перемещения узлов мобильной сети называется *random waypoint model*.

Модель передачи данных. Эта модель в ns2 задается следующими параметрами: [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]. В порядке перечисления: тип трафика (continuous bit rate или TCP); число узлов; случайное число, используемое для генерации случайной пары приемник — передатчик; максимальное число соединений между узлами; число передаваемых пакетов в секунду для каждого соединения. Размер пакета по умолчанию составляет 512 байт. Пары приемник—передатчик распределены по сети случайным образом.

Прочие параметры. К ним относятся:

- диаметр сети (максимальное расстояние между двумя разнесенными узлами сети);
- число узлов сети;
- время моделирования.

Набор тестов

Определенный набор тестов позволяет получать экспериментальные данные об определенных характеристиках сети; таким образом, постоянно добавляя новые тесты, можно добиться достаточно полного охвата всех основных характеристик.

На настоящий момент интерес представляют следующие тесты:

- работа сети в одной и той же конфигурации, но для разного числа узлов (набор № 1);
- работа сети в одной и той же конфигурации, но с разной степенью мобильности узлов (набор № 2);
- работа сети в одной и той же конфигурации, но с разной скоростью передачи данных (набор № 3).

Моделирование некоторых стеков протоколов

Ad-hoc сети отличаются тем, что такие вопросы как взаимодействие между всеми уровнями-протоколами, образующими стек, мобильность и масштабируемость, в них особенно важны какая бы задача не решалась [1, с. 25].

Рассмотрим с точки зрения поставленной задачи, какие именно протоколы могут быть промоделированы.

Подавляющее большинство существующих протоколов и их возможных сочетаний представлены в табл. 1.

Таблица 1

Стеки протоколов

№ стека	PHY/MAC уровень модели OSI	Сетевой уровень модели OSI	Транспортный уровень модели OSI
1	IEEE 802.11a/b/g	dsdv	TCP
2	IEEE 802.11a/b/g	aodv	TCP
3	IEEE 802.11a/b/g	dsr	TCP
4	IEEE 802.11a/b/g	tora	TCP
5	IEEE 802.11a/b/g	dsdv	ATP
6	IEEE 802.11a/b/g	aodv	ATP
7	IEEE 802.11a/b/g	dsr	ATP
8	IEEE 802.11a/b/g	tora	ATP
9	IEEE 802.11a/b/g	dsdv	NTCP
10	IEEE 802.11a/b/g	aodv	NTCP
11	IEEE 802.11a/b/g	dsr	NTCP
12	IEEE 802.11a/b/g	tora	NTCP
13	IEEE 802.15.4a	dsdv	TCP
14	IEEE 802.15.4a	aodv	TCP
15	IEEE 802.15.4a	dsr	TCP
16	IEEE 802.15.4a	tora	TCP
17	IEEE 802.15.4a	dsdv	ATP
18	IEEE 802.15.4a	aodv	ATP
19	IEEE 802.15.4a	dsr	ATP
20	IEEE 802.15.4a	tora	ATP
21	IEEE 802.15.4a	dsdv	NTCP
22	IEEE 802.15.4a	aodv	NTCP
23	IEEE 802.15.4a	dsr	NTCP
24	IEEE 802.15.4a	tora	NTCP

Таблица 2

Неизменяемые параметры моделирования

Параметры сети	Значение параметров
Максимальная скорость перемещения, м/с	10
Пауза между моментами движения, с	300
Тип трафика	Cbr
Скорость передачи, пакет/с	128
Скорость передачи, Кбит/с	512
Длина пакета, байт	512
Время моделирования, с	900
Диаметр сети, м	28

Анализ результатов моделирования

Параметры моделирования набора № 1 представлены в табл. 2 и 3.

Результаты моделирования набора № 1 представлены в табл. 4.

В табл. 4 представлены необработанные результаты, а именно: AGT, s — число посланных пакетов с пользовательскими данными; AGT, r — число принятых пакетов с пользовательскими данными; RTR, $s|f$ — число отправленных служебных пакетов.

Зависимости таких характеристик как PDF и NRL от числа узлов в сети при прочих неизменных параметрах представлены на рис. 1, 2.

На основании проведенных тестов можно сделать некоторые предварительные выводы, например, о том, какое оптимальное число узлов в сети соответствует каждому стеку с точки зрения надежности доставки. В нашем случае для стека № 1 это сеть с числом узлов не более 10. Для стека № 2 — не более 15. Таким образом, стек № 2 выигрывает по максимальному числу узлов в сети при прочем равном PDF. Промоделировав таким образом оставшиеся стеки, можно выбрать по критерию PDF наилучший, т. е. тот, который для сети численностью не более 20—25 узлов имеет минимальный уровень потери пакетов при доставке.

Таблица 3

Изменяемые параметры моделирования

№ теста	Число узлов	Макс. число соединений между узлами
1	5	2
2	10	3
3	15	5
4	20	7
5	25	9

Таблица 4

Результаты моделирования набора № 1 для стеков № 1 и № 2

№ стека	№ теста	AGT, s	AGT, r	RTR, $s\|f$
1	1	222666	222666	223018
1	2	319208	318723	321426
1	3	538647	458007	550518
1	4	765688	453146	791540
1	5	981939	428091	1029749
2	1	222843	222843	222853
2	2	319207	319196	356838
2	3	538830	470132	538905
2	4	765747	451204	793544
2	5	982240	412665	1047335

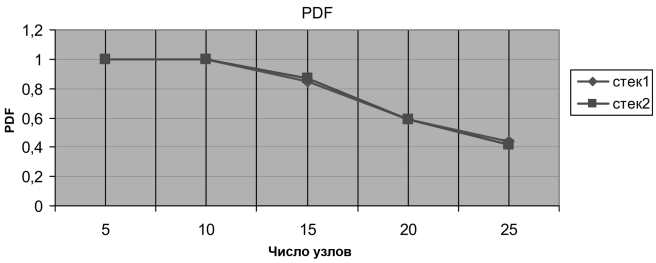


Рис. 1. Зависимость PDF от числа узлов

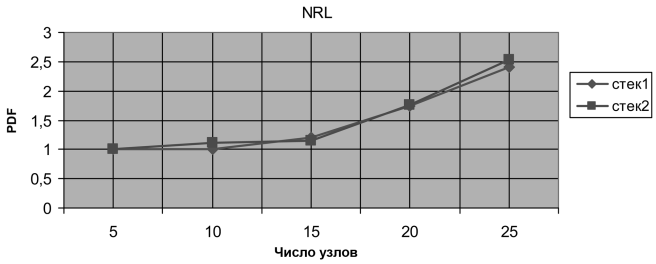


Рис. 2. Зависимость NRL от числа узлов

Последующая работа

В последующих работах необходимо промоделировать оставшиеся комбинации протоколов из табл. 1. Получив соответствующие экспериментальные оценки для всех комбинаций из табл. 1, появится возможность сравнения всех стеков протоколов по одному или более критериям из всего их множества. Новые протоколы или тесты могут быть легко добавлены в существующий набор, что позволит накопить в будущем статистику экспериментов и поможет создать некий справочник по основным параметрам ad-hoc сетей.

Список литературы

1. **AD HOC NETWORKS,** TECHNOLOGIES AND PROTOCOLS, edited by Prasant Mohapatra and Srikanth Krishnamurthy, 2005, Springer, print ISBN: 0-387-22689-3, eBooklSBN: 0-387-22690-7.

2. http://icawwwl.epfl.ch/uwb/ns-2/index.html

УДК 004.021

И. П. Норенков, д-р техн. наук, проф., зав. каф.,
В. А. Трудоношин, канд. техн. наук, доц.,
А. А. Кузьмин, аспирант,
И. А. Кузьмина, аспирант,
МГТУ им. Н. Э. Баумана

Генетические методы с фрагментными кроссовером и макромутациями

Приведены результаты экспериментального исследования эффективности генетических методов, основанных на фрагментном кроссовере и фрагментных макромутациях. Исследования проводились на тестовых задачах коммивояжера, компоновки и синтеза расписаний. Даны рекомендации по применению комбинированных генетических методов.

Ключевые слова: *генетический алгоритм, оптимизация.*

Введение. Генетические алгоритмы (ГА) применяют для решения сложных задач оптимизации и синтеза проектных решений, в том числе в условиях, когда альтернативные методы оказываются непригодными. Однако эффективность ГА, определяемая погрешностью получаемых решений и затратами вычислительных ресурсов, также не всегда удовлетворительна. Поэтому продолжается поиск новых ГА повышенной эффективности. Основное внимание при этом уделяется модификациям генетических операторов кроссовера, мутации и селекции.

Еще в начале активного развития ГА было обращено внимание на перспективность многоточечного кроссовера. В этом направлении следует отметить разработку алгоритма с однородным кроссовером [1], исследование поисковых и смешивающих возможностей многоточечного кроссовера [2, 3], объяснение эффективности многоточечного кроссовера на основе исследования вероятности разрыва шаблонов разных порядков [4] и ряд других результатов. Варианты оператора мутации рассматривались, например, в работе [5], в работе [6] описан алгоритм изменения вероятности мутации в процессе поиска экстремума. Детальное описание алгоритмов селекции приведено в работе [7].

Дополнительные резервы повышения эффективности генетических методов связаны с применением фрагментного кроссовера и фрагментных макромутаций [8].

В данной статье представлены результаты экспериментального исследования эффективности совместного использования фрагментных кроссовера и макромутаций с другими перспективными алгоритмами выполнения генетических операторов. Исследование проводилось на нескольких примерах характерных задач дискретной оптимизации. В качестве основного показателя эффективности генетического поиска принята точность, т. е. достигаемая степень приближения к наилучшему из известных результатов решения задачи, при заданном ограничении на время, затрачиваемое на получение решения.

Исследуемые генетические методы. К исследуемым методам относятся:

- фрагментный кроссовер;
- многохромосомная (*multiparent*) рекомбинация;
- фрагментные макромутации;
- фильтрация (*truncation*) особей при формировании новых поколений.

Далее различные комбинации исследуемых методов будут обозначаться двоичным вектором $\boldsymbol{X} = (x_1, x_2, x_3, x_4)$, где x_i — признак применения i-го метода.

Фрагментный кроссовер является разновидностью многоточечного кроссовера. Метод заключается в разделении родительских хромосом на фрагменты длиной L генов с выполнением операторов кроссовера и рекомбинации по отношению к каждому фрагменту. При $L = 1$ имеет место однородный (*uniform*) кроссовер, а при $L = n$ — одноточечный кроссовер, где n — длина хромосомы. При этом принято, что $x_1 = 1$ в случае фрагментного кроссовера с выбранным значением $L < n$; если применяется одноточечный кроссовер, то $x_1 = 0$.

Использование в ГА многохромосомной рекомбинации было предложено в работе [9]. В данном исследовании многохромосомная рекомбинация означает использование для каждого фрагмента своей пары родительских хромосом. При этом способ выбора родителей не критичен, так как он не является предметом исследования. Использовался пропорциональный отбор по правилу рулетки из имеющейся репродукционной группы. Признаком применения многохромосомной

рекомбинации является $x_2 = 1$. Если же применяется кроссовер с двумя родителями, т. е. с одной и той же парой родительских хромосом для всех фрагментов, то $x_2 = 0$.

Фрагментные макромутации выполняются с периодом в C поколений. Образуется репродукционная группа из $N/2$ хромосом текущего поколения, где N — размер популяции, каждая из хромосом этой группы порождает двух потомков. Аллели родительской особи сохраняются в нечетных фрагментах первого потомка и в четных фрагментах второго потомка. Гены остальных фрагментов потомков заполняются случайными значениями из области определения соответствующих аргументов. Важно отметить, что в новом цикле генетического поиска не сохраняются ранее достигнутые значения целевой функции (функции полезности), однако сохраняется генный состав популяции одновременно с внесением в него свежих элементов. Равномерное распределение мутированных генов по членам популяции обусловливает одинаковые шансы на выживание и развитие всех мутированных особей. Тем самым фрагментные макромутации способствуют преодолению преждевременной стагнации и выходу из локальных экстремумов с быстрым восстановлением ранее достигнутого уровня полезности и с существенными шансами на его превышение. В случае применения фрагментных макромутаций $x_3 = 1$, в противном случае $x_3 = 0$.

Фильтрация заключается в отбрасывании неперспективных особей, генерируемых в процессе кроссовера. Неперспективными являются хромосомы со значениями целевой функции выше плавающего порога, равного $(1 + h)Z_{\min}$, где $Z_{\min}$ — минимальное значение целевой функции, достигнутое на данном этапе генетического поиска (подразумевается минимизация целевой функции). Параметр h автоматически изменяется в процессе поиска, поддерживая порог на нужном уровне. В случае использования фильтрации хромосом $x_4 = 1$, иначе $x_4 = 0$. В репродукционную группу включается лучшая из двух возможных хромосом, генерируемых в каждом акте многоточечного кроссовера, причем при $x_4 = 1$ дополнительным условием для включения является преодоление целевой функцией плавающего порога.

Тестовые задачи. В качестве тестовых использовались *NP*-трудные задачи дискретной оптимизации:

- задача компоновки элементов в блоки (PP — *Partitioning Problem*);
- задача коммивояжера (TS — *Traveling Salesman*);
- многостадийная задача синтеза расписаний (SP — *Scheduling Problem*).

В задаче PP заданы множества элементов и блоков и матрица **D** межэлементных связей. Требуется распределить элементы по блокам с минимизацией числа Z межблочных связей и соблюдением ограничения $e_k \leqslant V$, где e_k — число элементов в k-м блоке, V — максимально допустимая загрузка одного блока, принято $V = 20$. К исходным данным относятся число элементов $n = 128$, число блоков $m = 8$. Решения PP кодировались следующим образом. Хромосома состоит из n генов, i-й ген соответствует i-му элементу, значением (аллелем) i-го гена является номер блока, в который этот элемент помещен.

В задаче TS заданы множество городов, представляемых в виде точек, и координаты каждого города, по которым определяется матрица $\mathbf{D} = [d_{ij}]$ расстояний между городами. В тестовой TS принято число городов $n = 120$. Хромосомы имеют длину, равную n. Первое правило кодирования решений задает следующий порядок включения городов в маршрут обхода городов: если $g_k < n$, где g_k — значение k-го гена, то g_k рассматривается как приоритет k-го города для включения в маршрут. Однако этого правила недостаточно для получения решений с достаточной точностью. Поэтому дополнительно использовалось второе правило, отражающее идеи метода комбинирования эвристик [10]: если $g_k \geqslant n$, то выбирается город с минимальным удалением от предыдущего города.

Содержанием задачи SP является составление расписания обслуживания n работ с их распределением во времени и по имеющимся машинам. Каждая работа проходит q стадий обслуживания, на каждой стадии для работы выбирается одна из m_k машин, $k = 1, \ldots q$, общее число машин равно M. Заданы матрица $\mathbf{P} = [p_{ij}]$, где p_{ij} — время обслуживания i-й работы на j-й машине. Работы сгруппированы в семейства и при переходе j-й машины с обслуживания работы одного семейства на обслуживание работы другого семейства выполняется переналадка машины. Заданы также времена и стоимости переналадок машин, цены одного часа работы каждой машины, временные ограничения на завершение обслуживания каждой работы. Требуется составить расписание минимальной стоимости Z при соблюдении временны́х ограничений. Исходные данные тестовой SP: $n = 105$, $M = 15$, $q = 4$. Для кодирования решений в SP использовался метод комбинирования эвристик. Эвристики различались правилами выбора работы и обслуживающей ее машины на очередном шаге синтеза расписания. Хромосома состояла из $nq = 420$ генов.

Результаты исследования. Расчеты проводились с ограничениями на время решения. В задаче SP достигнутое минимальное значение F целевой функции фиксировалось после 600 смен поколений, что соответствует приблизительно $T = 160$ тысячам вычислений целевой функции, длина

Результаты оптимизации

№	x_1	x_2	x_3	x_4	Целевая функция F			K_{ij}			Общая полезность K_{fit} метода
					SP	TS	PP	SP	TS	PP	
1	1	1	1	1	21818	549	110	1	1	1	1
2	1	0	1	1	21845	554	116	0,91	0,81	0,74	0,82
3	1	1	1	0	21933	557	114	0,61	0,69	0,83	0,71
4	1	0	1	0	21945	556	117	0,57	0,70	0,74	0,67
5	1	1	0	1	21904	557	119	0,71	0,61	0,61	0,64
6	1	0	0	1	21954	557	118	0,54	0,69	0,65	0,63
7	1	0	0	0	22042	560	120	0,25	0,58	0,57	0,47
8	1	1	0	0	22024	557	124	0,31	0,69	0,39	0,46
9	0	0	1	0	22053	562	123	0,21	0,50	0,43	0,38
10	0	0	0	0	22038	567	125	0,26	0,31	0,35	0,31
11	0	0	1	1	22043	566	133	0,24	0,35	0	0,20
12	0	0	0	1	22116	575	120	0	0	0,57	0,19

цикла C принималась равной 100 поколениям, размер популяции $N = 100$, длина фрагментов $L = 7$.

В задаче TS расчеты заканчивались после выполнения 200 тысяч оценок целевой функции, размер популяции $N = 60$, $C = 150$, $L = 5$.

В задаче PP были приняты следующие данные: длительность расчетов $T = 300\,000$, $N = 100$, $C = 150$, $L = 4$.

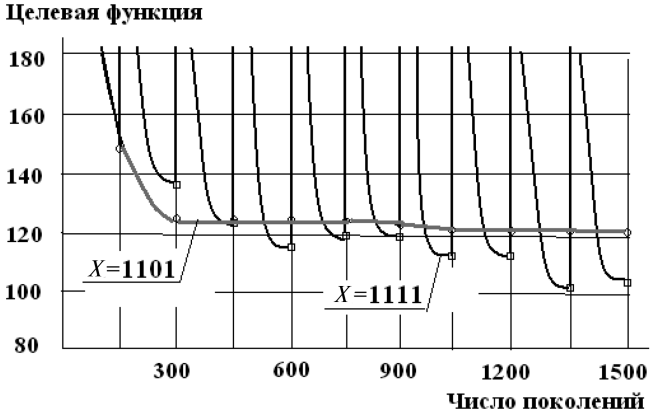
Результаты оптимизации (усредненные по нескольким вариантам расчета каждой задачи) приведены в таблице. Каждое сочетание методов (далее называемое комбинированным генетическим методом КГМ) отмечено в таблице указанием его кода $\boldsymbol{X}$. В колонках 6—8 даны полученные оценки F целевой функции для каждой задачи и для каждого КГМ. По этим значениям устанавливались нормированные показатели полезности методов для задач SP, TS, PP — коэффициенты K_{ij}, рассчитываемые по формуле

$$K_{ij} = (F_{\max i} - F_{ij})/(F_{\max i} - F_{\min i}),$$

где $F_{\max i}$, $F_{\min i}$ и F_{ij} — максимальное, минимальное и полученное с помощью j-го метода значение целевой функции в i-й задаче, $i = 1, 2, 3$. Общая полезность метода оценивалась усреднением показателей полезности по трем задачам. Полученные таким образом значения общей полезности КГМ в виде значений коэффициента K_{fit} приведены в последнем столбце таблицы.

Строки таблицы упорядочены по значениям полезности K_{fit}.

На рисунке приведены сглаженные графики изменения целевой функции в одном из вариантов решения задачи PP при $C = 150$ в процессе поиска экстремума без применения ($\boldsymbol{X} = 1101$) и с применением фрагментных макромутаций ($\boldsymbol{X} = 1111$), из которых видно лучшее приближение к экстремуму в случае $\boldsymbol{X} = 1111$.



Сравнение траекторий поиска в КГМ с применением и без применения фрагментных макромутаций

Обсуждение результатов. Как видно из данных таблицы, фрагментный кроссовер и фрагментные макромутации при их совместном применении обеспечивают наибольшую эффективность генетического поиска. Их раздельное применение менее эффективно. Наихудшие значения общей полезности имеют методы без использования фрагментного кроссовера. В большинстве случаев замена в коде $\boldsymbol{X}$ значения x_i с 0 на 1 вызывает увеличение усредненной полезности K_{fit} комбинированных методов. Следовательно, фрагментный кроссовер, фрагментные макромутации и в сочетании с фрагментным кроссовером многохромосомная рекомбинация и фильтрация способствуют повышению эффективности генетических алгоритмов, хотя в отдельных задачах и вариантах расчета возможны отступления от этой общей тенденции.

Лучшим сочетанием методов оказался КГМ с кодом $\boldsymbol{X} = 1111$, в котором используются все четыре исследуемых алгоритма.

Список литературы

1. **Syswerda G.** Uniform Crossover in Genetic Algorithms // Proc. 3rd Int. Conf. on Genetic Algorithms, Morgan Kaufmann Publ., 1989.

2. **Prugel-Bennett A.** The mixing rate of different crossover operators // Foundations of Genetic Algoritrims. 2001. N 6.
3. **Sastry K., Goldberg D., Kendall G.** Genetic Algorithms / — http://citeseer.ist.psu.edu/cache/papers/cs2/258/.
4. **De Jong K. A., Spears W. M.** A Formal Analysis of the Role of Multi-point Crossover in Genetic Algorithms. Annals of Mathematics and Artificial Intelligence, 1992.
5. **Back T.** Optimal Mutation Rates in Genetic Search // Proc. 5th Int. Conf. on Genetic Algorithms. Morgan Kaufmann Publ., 1993.
6. **Fogarty T.** Varying the Probability of Mutation in Genetic Algorithm // Proc. 3rd Int. Conf. on Genetic Algorithms. Morgan Kaufmann Publ., 1989.
7. **Blickle T., Thiele L.** A Comparison of Selection Schemes used in Genetic Algorithms. TIK Report, http://qai.narod.ru/Papers/blickle_95.pdf.
8. **Норенков И. П.** Исследование эффективности генетического метода с фрагментным кроссовером // Информационные технологии. 2008. № 6. С. 26—29.
9. **Eiben A. E., Raue P.-E., Ruttkay Zs.** Genetic Algorithms with Multi-parent Recombination // In Proc. of the 3d Conf. on Parallel Problem Solving from Nature. Springer-Verlag, 1994.
10. **Норенков И. П.** Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.

УДК 004.932.001.57

Т. Г. Кязимов, канд. физ.-мат. наук,
доцент, зав. отделом,
Ш. Д. Махмудова, зав. сектором,
Институт Информационных Технологий
Национальной Академии Наук Азербайджана,
г. Баку

Система компьютерного распознавания людей по фотопортретам

Рассматривается методика поиска человека в базе изображений по его фотопортрету. На основе выбранных идентификационных точек лица вычисляются расстояния между ними. Идентификационные признаки лица определяются способом, принципиально отличным от ранее известных.

Ключевые слова: *идентификация, база изображений, антропометрические точки, признаки, ключевые признаки.*

Введение

Человеческое зрение, в отличие от зрения других живых существ, наделено качественной способностью быстрого узнавания знакомых объектов, вещей, людей и многого другого. Человек может распознавать предметы, вещи, животных и других людей только по их изображениям, не используя иные органы чувств, например обоняние или слух.

Как можно заметить, идентификация человека человеком осуществляется почти мгновенно (моментально) по разным признакам, таким как запах, голос, одежда и т. д. Однако изображение лица является ключевым признаком при распознавании человека.

Проблема формализации процесса распознавания человеческих лиц рассматривалась еще на заре развития систем распознавания образов и до сих пор остается актуальной. Но последние десятилетия количество научных исследований и публикаций увеличилось в несколько раз, что и свидетельствует о возрастающей актуальности данной проблемы. Это объясняется в первую очередь возрастающими возможностями компьютерной техники и удешевлением ее эксплуатации. Но вместе с тем повышенное внимание к биометрическим технологиям диктуется и существованием обширного круга коммерческих и социальных задач, где автоматическая идентификация человека является неотъемлемой частью их успешного применения. Так, например, идентификация человека по изображению его лица может применяться в системах контроля удостоверений личности (паспорта, водительских прав), информационной безопасности (доступ к компьютерам, базам данных и т. д.), наблюдения и расследования криминальных событий, а также в банковской сфере (банкоматах, системах удаленного управления счетом) [1, 2].

К настоящему времени имеется значительное число работ, посвященных исследованиям распознавания людей по фотопортретам, а также некоторые рекомендации разработчикам систем идентификации личности по фотографии [3—13]. При этом под термином "фотопортрет" подразумевается цифровое изображение лица человека в фас без элементов одежды, украшений, солнечных очков и т. д., которые могут закрывать или искажать части лица.

Очевидно, что люди существенно отличаются друг от друга такими чертами лица, как расположение глаз, бровей, носа, ушей, рта и т. д. Поэтому не удивительно, что исторически первый подход к решению проблемы автоматической идентификации человека по изображению его лица был основан на выделении и сравнении некоторых антропометрических характеристик лица. Этот метод давно используется в практической криминалистике, однако замеры и сравнения выполнялись вручную. Он особенно эф-

фективен в случае, когда у человека нет других фотографий, кроме фотографии в документе (документный контроль). Основной проблемой данного подхода является выбор и определение совокупности характерных точек человеческого лица, по которым будет осуществляться идентификация. Однако при этом должны быть учтены некоторые требования, предъявляемые к портретам:

- идентификационные точки не должны закрываться прической, бородой, маской и т. п.;
- процесс распознавания не должен зависеть от масштаба изображения;
- система идентификационных точек должна обеспечивать относительную устойчивость процесса распознавания при незначительном изменении ракурса съемки (легкий поворот головы, наклон, изменение выражения лица и т. д.);
- число характерных точек должно быть минимальным для обеспечения высокой точности распознавания.

В литературе имеется большое число работ, посвященных решению различных аспектов этой проблемы [3, 5—13].

В данной работе рассматривается методика поиска человека в базе данных изображений по заданному цифровому портрету, которая основана на специально разработанных геометрических характеристиках лица.

Выделение особых точек

Как показывает криминалистическая практика, необходимо выделить около 30 особых точек на изображении человека. Эти точки должны быть максимально устойчивыми к небольшим изменениям (ракурса, освещения, мимики, косметики, возрастных изменений и т. п.) изображения.

В процессе предварительных экспериментов были отобраны 19 особых точек лица, которые показаны на рис. 1.

Как следует из рис. 1 идентификационные точки обозначены следующим образом: центр брови (1а и 1в); центр зрачка (2а и 2в); верхние крайние точки ушей (3а и 3в); правый угол правого глаза — 4а; левый угол левого глаза — 4в; левый угол правого глаза — 5а; правый угол левого глаза — 5в; нижние точки окончания мочек ушей (6а и 6в); крайние точки носа по горизонтали (7а и 7в); кончик носа (8), который определяется как центральная точка между носовыми отверстиями; уголки рта (9а и 9в); центр рта (10) — как точка пересечения линии, разделяющей верхнюю и нижнюю губы объекта, и перпендикуляра, опущенного из точки, определяющей кончик носа объекта; кончик подбородка (11).

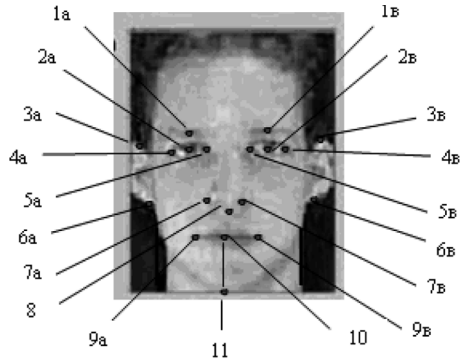


Рис. 1. Антропометрические точки на фронтальной проекции лица человека

Будем выделять следующие расстояния (рис. 2):

1) между центрами сетчатки глаз (2а, 2в);

2) между внутренними уголками глаз (5а, 5в);

3) между центром сетчатки глаза и центром брови [(1а, 2а), (1в, 2в)];

4) между центром сетчатки глаза и серединой линии смыкания губ [(2а, 10), (2в, 10)];

5) между центром сетчатки глаза и нижней точкой носа [(2а,8), (2в, 8)];

6) максимальная ширина носа (7а, 7в);

7) между центром сетчатки глаза и подбородком [(2а, 11), (2в, 11)];

8) между серединой линии смыкания губ и подбородком (10, 11);

9) между кончиком носа и подбородком (8, 11);

10) ширина рта (9а, 9в);

11) ширина лица на уровне линии глаз;

12) ширина лица на уровне нижней точки носа;

13) ширина лица на уровне линии смыкания губ;

14) между наружным уголком глаза и верхней точкой уха [(3а, 4а), (3в, 4в)];

15) между верхними точками ушей (3а, 3в);

16) между нижними точками ушей (6а, 6в);

17) между верхней и нижней точками уха [(3а, 6а), (3в, 6в)].

Расстояния (1); (2), (4), (5), (6), (7), (8), (11) будем считать основными, поскольку влияние на них таких факторов, как прическа, макияж, украшения и др. незначительны.

В имеющихся в этой области работах [4—7] для определения признаков, т. е. соотношения рас-

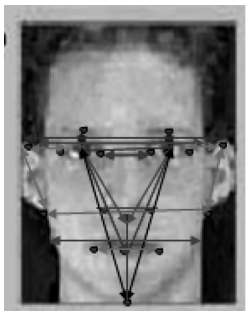


Рис. 2. Расстояния между антропометрическими точками

стояний, использовались или всевозможные соотношения, или же некоторые выбранные соотношения имеющихся расстояний. Нами предлагается вариант вычисления признаков, несколько отличный от ранее уже использованных признаков. Разъясним суть этого отличия.

Пусть имеется множество расстояний $S(s_i \in S, i = \overline{1, n})$. Элемент p_i множества признаков $P(p_i \in P, i = 1, n - 1)$ определяем следующим образом:

$$p_i = \frac{s_i}{s_{i+1}}, \quad i = \overline{1, n-1}.$$

Множество P будем считать базисным множеством признаков. Докажем, что любое множество признаков, построенное на основе соотношений элементов множества S, можно получить из элементов P путем конечного числа арифметических операций. Введем следующие обозначения:

$$p_{kj} = \frac{s_k}{s_j}, \text{ где } k, j = \overline{1, n}; \quad k \neq j, \ | k - j| \neq 1, \quad k < j.$$

Лемма. Если известны P_i $(i = \overline{1, n})$, то

$$p_{kj} = \prod_{l=k}^{j} p_l.$$

Действительно, если раскрыть произведение, то получим

$$\prod_{l=k}^{j-1} p_l = \frac{s_k}{s_{k+1}} \cdot \frac{s_{k+1}}{s_{k+2}} \ldots \frac{s_{j-2}}{s_{j-1}} \cdot \frac{s_{j-1}}{s_j} = \frac{s_k}{s_j} = p_{kj}.$$

Суть доказанного заключается в том, что базисное множество уже содержит информацию о других соотношениях расстояний и, следовательно, нет необходимости использования их в качестве признаков. Другими словами, рассмотрение базисного множества для идентификации можно считать достаточным. Следует отметить, что в случае, когда признаки вычисляются путем деления всех расстояний на расстояние между зрачками, полученное множество тоже является базисным множеством.

Введение признаков в виде отношения для идентификационных единиц делает их масштабно не зависящими от расстояния, с которого снимается фотография человека. В данном случае использование реальных размеров головы и ее участков невозможно определить, а для признаков совершенно неважно, на каком расстоянии находился человек во время съемки от объектива.

Дополнительно для практических целей расстояния (1)—(17) были разделены на две группы:

- расстояния, измеряемые в горизонтальном направлении ((1), (2), (6), (10), (11), (12), (13), (14), (15), (16));
- расстояния, измеряемые в вертикальном направлении.

Несомненно, признаки, составленные на основе соотношений между расстояниями, входящими в первую группу, будут достаточно устойчивыми при повороте головы человека по вертикальной оси фотографии, а признаки, составленные на основе расстояний второй группы, будут достаточно устойчивыми наклону головы человека вниз или вверх по горизонтальной оси. Считаем, что такая же устойчивость признаков будет сохранена в случае одновременного поворота и наклона головы человека. Пределы поворота и наклона головы человека на фотографии, конечно же, будут определены возможностями выделения особых точек и определения соответствующих расстояний.

Эксперименты показали достаточно хорошие результаты (около 1— 1,5 % отклонений) по устойчивости признаков в группах при повороте головы человека до 25° и наклоне — до 15°. Отклонение головы влево или вправо не учитывалось.

Организация системы идентификации личности на основе антропометрических точек лица

При организации системы идентификации на основе антропометрических точек лица особую роль играют способы формирования баз данных изображений. Не перечисляя известные способы организации базы и проведения в ней поиска, а также сравнения хранящихся в ней данных, перейдем к описанию базы данных изображений разработанной авторами системы идентификации личности на основе указанных выше принятий и рассуждений.

База данных формируется на основе данных, полученных из отдела кадров предприятия, и изображения личности. На момент пополнения базы данных для этой личности особые точки (см. рис. 1) определяются вручную и одновременно в двух группах, автоматически определяются и хранятся расстояния (1)—(17). Далее вычисляются признаки P для соответствующих групп и тоже хранятся в базе. Эти данные определяются и вычисляются лишь один раз в момент формирования базы данных.

Определенные данные человека (пол, раса, возраст, регион, особые приметы и т. п.), которые имеются в базе данных, могут служить ключом поиска.

Задача идентификации сводится к нахождению из базы данных нескольких изображений (от одного до десятка), наиболее похожих на заданное.

Заданное изображение сравнивается с изображениями, имеющимися в базе данных, путем вычисления евклидового расстояния между двумя точками в 16-мерном пространстве:

$$S_j(P_i^*, P_i^j) = \sqrt{\sum_{i=1}^{n-1} (P_i^* - P_i^j)^2},$$

$$(i = \overline{1, n-1}, \quad j = \overline{1, N}),$$

где $P_i^*(i = 1,\ n - 1)$ — параметры изображения идентифицируемого человека; $P_i^j(i = 1,\ n - 1,\ j = \overline{1, N})$ — параметры изображения j-го человека в базе данных.

Используя указанные выше ключи поиска, можно значительно уменьшить число проверяемых изображений (портретов).

Выводы

Разработанная по предложенному методу система может применяться в системах контроля удостоверений личности (паспорта, водительских прав), информационной безопасности (доступ к компьютерам, базам данных и т. д.), наблюдения и расследования криминальных событий, а также в банковской сфере.

Список литературы

1. **Горелик А. Л., Скрипкин В. А.** Методы распознавания. М.: Высшая школа, 2004.
2. **Зинин А. М., Кирсанова Л. З.** Криминалистическая фотопортретная экспертиза. М.: Наука, 1991.
3. **Средства** контроля доступа // Иностранная печать о техническом оснащении полиции капиталистических государств. М.: ВИНИТИ. 1992. № 4. С. 12—27.
4. **Самаль Д. И., Старовойтов В. В.** Подходы и методы распознавания людей по фотопортретам. Минск, 1998. 54 с. (Препринт / Ин-т техн. кибернетики НАН Беларуси; № 8).
5. **Старовойтов В. В.** Локальные геометрические методы цифровой обработки и анализа изображений. — Минск: Изд. Ин-та техн. кибернетики НАН Беларуси, 1997. 284 с.
6. **Снетков В. А., Виниченко И. Ф., Житников В. С. и др.** Криминалистическое описание внешности человека. М.: Изд. МВД СССР ВНИИ, 1984.
7. **Starovoitov V., Samal D., Votsis G., Kolias S.** Geometric features for face recognition // Proc. PRIP'99, Minsk, Belarus, May 18—20, 1999.
8. **Achermann B., Bunke H.** Combination of face classifiers for person identification // Proc. ICPR, 1996. Vol. 4. P. 416—420.
9. **Abay E., Akarum L., Alpaydyn E.** A comparative analysis of different feature sets for face recognition // Proc. ISCIS, Antalya, 1997.
10. **Brunelli R., Poggio T.** Face recognition: features versus templates // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993. Vol. 15. N 10. P. 1042—1052.
11. **Cox I. J., Ghosn J., Yianilos P. N.** Feature-based face recognition using mixture distance // NEC Research Institute, Technical Report #95-09, 1995.
12. **Kanade T**. Picture processing by computer complex and recognition of human faces // PhD thesis, Kyoto University, 1973.
13. **Lawrence S., Giles C. L., Tsoi A. C., Back A. D.** Face recognition: a convolutional neural network approach // IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition, 1997.

УДК 004.8; 004.82

И. Л. Артемьева, канд. техн. наук, ст. научн. сотр.,
Институт автоматики и процессов управления
ДВО РАН, г. Владивосток

Сложно структурированные предметные области. Построение многоуровневых онтологий[1]

В настоящее время общепризнана важность использования онтологий как основы для спецификации и разработки программного обеспечения, поддержки общего доступа к информации, разработки порталов знаний, пользовательского интерфейса программных систем и редакторов информации. Однако существующие определения онтологий и методологии их создания не охватывают случай сложно структурированных предметных областей, т. е. таких областей, разделы которых имеют разные, но похожие онтологии, подразделы разделов, в свою очередь, имеют разные, но похожие онтологии и т. д. В статье определены класс сложно структурированных предметных областей и структура их многоуровневых онтологий, а также описаны методы разработки онтологий таких областей.

Ключевые слова: *онтология предметной области, сложно структурированная область, разработка онтологии для сложно структурированной области.*

Введение

В настоящее время отмечается важность использования онтологии как основы для спецификации и разработки программного обеспечения [1], поддержки общего доступа к информации, поиска информации, поддержки взаимодействия при объединении информации, создания порталов знаний [2—4], разработки пользовательского интерфейса программных систем [5] и редакторов информации [6—7]. Целью многих исследований является разработка точных, формальных каталогов знаний, которые могут использоваться интеллектуальными системами. Под онтологией во многих работах понимается формальное явное описание понятий (часто называемых классами) в предметной области, свойств (иногда называемых слотами) каждого понятия, задающее различные особенности и атрибуты понятия, а также ограничения на свойства (иногда называемых фасетами) [8—9]. Онтология определяет термины, используемые для описания и представления области

[1] Работа выполнена при финансовой поддержке ДВО РАН в рамках Программы Президиума РАН № 14 "Фундаментальные проблемы информатики и информационных технологий", проект 06-1-14-051 "Интеллектуальные системы, основанные на многоуровневых моделях предметных областей".

знаний. Онтология вместе с набором индивидуальных случаев классов составляет базу знаний.

В настоящее время разработаны различные методологии создания онтологий [1, 10—12]. Во многих методологиях разработка онтологии включает:

- определение классов в онтологии;
- расположение классов в таксономическую иерархию (подкласс — надкласс);
- определение слотов и описание допускаемых значений этих слотов;
- заполнение значений слотов экземпляров [13].

С использованием данных методологий разработаны различные онтологии (см., например, http://musing.deri.at/ontologies/v0.3/ и http://www.daml.org/ontologies/).

Однако существует ряд нерешенных проблем. Предлагаемые определения онтологий не охватывают случай сложно структурированных предметных областей (ПО), т. е. областей, разделы которых имеют разные, но похожие онтологии [15], подразделы разделов которых, в свою очередь, имеют разные, но похожие онтологии. Top-level и upper-level онтологии [14] содержат определения философских понятий, т. е. понятий высокого уровня абстракции, что затрудняет их использование при моделировании предметных областей. Часто уровень, понимаемый как уровень онтологии, является характеристикой понятия онтологии, определяющий уровень класса в иерархии классов.

Целью данной работы является определение устройства онтологий для сложно структурированных областей и описание методов создания таких онтологий.

Определение класса сложно структурированных предметных областей

Назовем предметную область сложно структурированной, если она обладает следующими свойствами:

- в ней существуют разделы, которые описываются в разных, но похожих [15] системах понятий;
- разделы, в свою очередь, имеют подразделы, которые описываются в разных, но похожих системах понятий;
- любой подраздел, в свою очередь, может иметь подразделы, обладающие указанным свойством, и т. д.

Раздел (и подраздел) сложно структурированной ПО является также предметной областью, в которой происходит своя профессиональная деятельность и которая характеризуется своим множеством задач, причем среди множества задач разных разделов могут существовать похожие задачи. При решении задач профессиональной деятельности в сложно структурированной ПО могут использоваться понятия онтологии ее разных разделов, а также знания разных разделов.

Примером сложно структурированной ПО является медицина [16]. В данной области примерами разделов являются терапия, хирургия и др. Терминами онтологии каждого раздела являются названия заболеваний, изучаемых в данном разделе, а также названия признаков, значения которых используются при решении задачи диагностики заболеваний специалистами раздела. Множества названий заболеваний и признаков в каждом разделе свои.

Другим примером сложно структурированной ПО является химия [17]. Примерами ее разделов являются физическая, органическая и аналитическая химия. Физическая химия изучает физико-химические процессы [18]. Описание этих процессов дается в терминах свойств участвующих в процессах веществ и реакций. Органическая химия добавляет терминологию, позволяющую говорить о структурных свойствах веществ [19—20]. Аналитическая химия изучает процессы воздействия на вещества различными излучениями [21]. Примерами подразделов для физической химии являются химическая термодинамика и химическая кинетика, для аналитической химии подраздел связан с конкретным методом анализа (например, рентгено-флуорисцентный анализ).

Еще одним примером сложно структурированной ПО является область "Преобразования программ". В этой области изучаются процессы изменения программ в результате применения различных преобразований [22]. Примерами разделов являются "Преобразования структурных программ", "Преобразования параллельных программ". Описание преобразований дается в терминах свойств языков, на которых записаны программы.

При создании интеллектуальных систем для сложно структурированных ПО существует проблема интеграции знаний разных разделов и подразделов в рамках одной базы знаний. Средством для такой интеграции является онтология. Но она должна учитывать, что системы понятий (онтологии разделов или подразделов), используемые в разных разделах и подразделах, отличаются друг от друга. Поэтому второй проблемой является способ интеграции уже этих систем понятий (онтологий). Средством решения данной проблемы может быть использование онтологий более высокого уровня общности.

Структура онтологии предметной области

Онтология предметной области состоит из онтологии действительности, онтологии знаний и соглашений, задающих связи этих двух онтологий [23]. *Онтология знаний* определяет термины, исполь-

Рис. 1. Структура онтологии предметной области "Химия"

зуемые при задании знаний предметной области, и ограничения целостности знаний. *Онтология действительности* определяет термины, используемые при задании исходных данных задач, решаемых программными системами. В терминах онтологии действительности представляются результаты решения задач.

Так, для предметной области "Химия" (рис. 1) термины онтологии действительности позволяют описывать различные свойства физико-химических процессов, а термины антологии знаний — свойства химических элементов, веществ, реакций, элементов в составе вещества, веществ как участников реакций и т. д.

Свойства онтологий сложно структурированных предметных областей

Для описания свойств онтологий сложно структурированных предметных областей рассмотрим примеры. В онтологии знаний физической химии [18] можно выделить термины, которые представляют собой названия свойств химических элементов. Значением каждого такого термина является отображение, областью определения которого является множество химических элементов, а областью значений — множество значений свойства. Такие же термины можно найти и в онтологии органической химии и онтологиях других разделов данной области. Кроме терминов, которые определяют названия свойств элементов, существуют также термины, которые представляют собой названия свойств химических соединений, реакций, радикалов и т. д.

Анализ онтологий разных разделов химии [18—21] показал, что существуют и другие множества терминов, которые обладают некоторыми общими свойствами. Такими являются термины, позволяющие определить состав химических элементов или химических веществ. Каждый такой термин представляет собой функцию, аргументом которой является химический элемент или вещество, а результатом — множество компонентов элемента или вещества. Существуют также термины, позволяющие определить различные свойства элемента в составе вещества, вещества как участника реакции и т. д. Существует также сходство между онтологиями действительности разных разделов химии. Физико-химический процесс, изучаемый физической химией, и химический процесс, изучаемый органической химией, состоят из последовательности шагов. Участниками процесса являются химические вещества. В ходе процесса изменяются свойства этих веществ либо в результате физических воздействий, либо в результате химических реакций.

Теперь введем понятие *уровней общности* [24]. Множество вербальных представлений информации о действительности предметной области имеет уровень 0.

Онтология вместе с базой знаний имеет уровень 1. Заметим, что на уровне 1 терминам онтологии действительности не сопоставлены значения. Сопоставляя терминам онтологии действительности конкретные значения, получим вербальное представление информации о конкретной ситуации, которое будет принадлежать уровню 0.

Онтология предметной области без системы знаний имеет уровень 2. Задавая разные базы знаний, не противоречащие онтологии, будем получать разные онтологии с системой знаний уровня 1.

Онтология, которая будет описывать свойство множества онтологий уровня 2, уже будет иметь уровень 3. Ее термины определяют множества терминов онтологии уровня 2, задавая свойства элементов этих множеств. Например, для рассмотренных примеров онтология уровня 3 будет содержать следующие термины:

- "собственные свойства элементов" — определяет свойства множества терминов, обозначающих отображения, областью определения которых является множество химических элементов, а область значений зависит от термина, задающего название свойства;
- "собственные свойства веществ" — определяет свойства множества терминов, обозначающих отображения, областью определения которых является множество химических веществ, а область значений зависит от термина, задающего название свойства;
- "компоненты вещества" — определяет свойства множества терминов, обозначающих отображения, областью определения которых является множество химических веществ, а область значений зависит от типа компонента: это мо-

жет быть, например, множество химических элементов и т. д. [17].

Задание свойства множества терминов онтологии уровня 2 состоит либо в определении объема каждого понятия, обозначенного термином, либо в определении способа вычисления этого объема в зависимости от значений параметров, связанных с множеством определяемых терминов. В первом случае онтология уровня 3 содержит термины-имена множеств терминов онтологии уровня 2, а также утверждения, определяющие объемы понятий элементов каждого множества. Во втором случае онтология определяет функции, аргументами которых являются параметры, от которых зависит объем понятий, обозначаемых терминами — элементами множеств.

Задание терминов онтологии уровня 2 в первом случае состоит в перечислении элементов всех множеств, имена которых определены в онтологии уровня 3. Во втором случае для понятия онтологии уровня 2 определяется его название, а объем задается применением функции к конкретному набору значений параметров. Отметим, что для сложно структурированной предметной области множества терминов разных онтологий уровня 2 будут различными, т. е. различными будут множества терминов для описания действительности и знаний.

Из рассмотренного выше примера видно, что существует сходство уже между онтологиями уровня 3, что позволяет определить онтологию следующего уровня, которая будет описывать свойство множества онтологий уровня 3.

Термины онтологии более высокого уровня будут задавать имена множеств понятий. Для рассмотренного примера такими терминами могут быть термины "собственные свойства сущностей" (множество функций, областью определения которых является множество сущностей некоторого типа, а область значений зависит от термина, задающего название свойства) [17]. В этом случае также появляются термины, обозначающие вспомогательные понятия. В рассмотренном примере таким термином является термин "типы сущностей", который обозначает множество названий типов сущностей (его значением могут быть термины "химические элементы", "химические вещества" и т. д.).

Таким образом, для сложно структурированной предметной области онтология уровня $i + 1$ описывает свойство множества онтологий уровня i. Онтологию верхнего уровня будем называть онтологией предметной области (рис. 2). Она содержит термины, с помощью которых определяются онтологии следующего уровня (онтологии разделов). Эти онтологии образуют множество онтологий (множество модулей) разделов предметной области. Переход к онтологии некоторого раздела

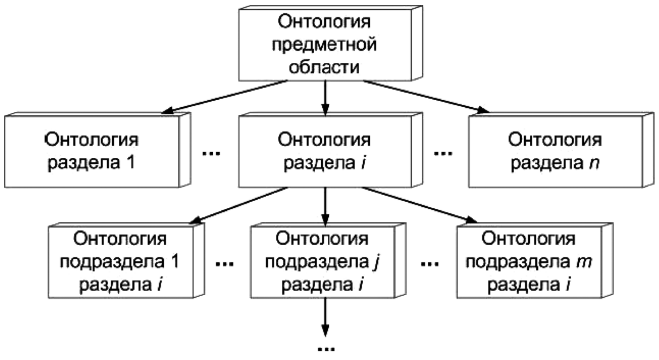


Рис. 2. Структура онтологии сложно структурированной области

от онтологии предметной области состоит в задании терминов онтологии этого раздела, а также онтологических соглашений.

Онтология каждого раздела содержит термины, с помощью которых определяется онтология подраздела. Переход к онтологии подраздела состоит в задании терминов онтологии этого подраздела, а также онтологических соглашений. Таким образом, онтологии каждого раздела соответствует множество онтологий подразделов данного раздела. Если подраздел, в свою очередь, имеет подразделы, то его онтология используется при определении онтологий подразделов.

Знания предметной области также состоят из модулей. Модуль знаний формулируется в терминах соответствующей онтологии подраздела и содержит знания этого подраздела.

Методы разработки онтологий для сложно структурированных областей

При выполнении анализа "снизу вверх" для сложно структурированной ПО предполагается, что онтологии уровня 2 для нескольких разделов уже построены с использованием какого-либо из существующих методов анализа. Задачей анализа "снизу вверх" является поиск "регулярностей" в этих онтологиях.

Для построения онтологии уровня 2 множество онтологических соглашений онтологии уровня 2 разбивается на группы таким образом, чтобы в одну группу попали соглашения с похожим смыслом. Дается формулировка общего смысла для соглашений каждой группы и определяются термины онтологии уровня 3, позволяющие этот смысл представить. Для каждого термина задается область значений. Этот шаг повторяется для всех групп соглашений.

Далее выполняется разбиение множества терминов онтологии уровня 2 на группы таким образом, чтобы в одну группу попали термины с похожим смыслом или с похожей схемой определения. Для каждой группы определяются термин, обозначающий название группы, область значе-

ний термина, а также способ определения областей значений всех терминов группы. Этот шаг повторяется для всех введенных групп.

В заключение формулируются онтологические соглашения, задающие ограничения на множества значений введенных терминов и связи между значениями терминов.

Введенные на данном этапе термины имеют смысл параметров онтологии уровня 3: их значения либо позволяют определить термины онтологий уровня 2, либо онтологические соглашения этих онтологий.

Построение онтологии уровня i выполняется в соответствии с приведенной схемой анализа, если построены несколько онтологий уровня $i-1$, т. е. в этом случае происходит поиск "регулярностей" в онтологиях уровня $i-1$.

Теперь определим метод анализа "сверху вниз". Пусть построена онтология уровня m. Эта онтология задает схему анализа раздела предметной области при построении онтологии уровня $m-1$. Определим этапы анализа методом "сверху вниз".

1. Вначале должны быть определены понятия рассматриваемой области, которые могут быть значениями параметров онтологии уровня m. Для каждого такого понятия необходимо проверить, совпадает ли его объем с объемом понятия, определяемого онтологией уровня m.

2. После того как значения параметров онтологии уровня m определены, проверяется, все ли множество онтологических соглашений уровня $m-1$ определяет онтология уровня m при заданных значениях параметров. Для этого выполняется подстановка заданных значений терминов во множество онтологических соглашений уровня m, в результате которой будет построено множество онтологических соглашений уровня $m-1$. Если онтология уровня m определяет не все онтологические соглашения уровня $m-1$, то формулируются недостающие соглашения в терминах онтологии уровня $m-1$. Они задают дополнительные ограничения на значения терминов онтологии уровня $m-1$.

3. Определяются термины — элементы множеств, определение которых зависит от параметра. При этом также необходимо убедиться, что объемы понятий, обозначенных такими терминами, совпадают с объемами понятий, заданными при определении в онтологии уровня m множеств, зависящих от параметра.

4. Определяется подмножество терминов, которые играют роль параметров уровня $m-1$. Эти термины используются в дальнейшем на первом шаге анализа "сверху вниз" при построении онтологии уровня $m-2$.

Если оказалось что онтология высокого уровня не позволяет задать все термины и онтологические соглашения определяемой онтологии, то проводится ее усложнение. Для этого выполняется анализ "снизу вверх", в результате которого добавляются новые термины и онтологические соглашения в онтологию верхнего уровня.

Построенная онтология уровня $m-1$ определяет схему анализа подразделов данного раздела предметной области в целях разработки их онтологий. Выполняются этапы 1—4 для каждого подраздела.

Схему анализа знаний задает модуль онтологии уровня 2. Вначале определяются значения параметров онтологии уровня 2, т. е. определяются структурированные знания в соответствии со структурой, задаваемой определениями параметров онтологии уровня 2. Далее формулируются знания, для которых онтология уровня 2 не определяет структуру.

Метод "снизу вверх" использован при создании четырехуровневой онтологии химии, а метод "сверху вниз" — при разработке онтологии раздела "Катализ", а также при развитии онтологии рентгенофлуоресцентного анализа.

Список литературы

1. **Клещев А. С.** Использование онтологий в разработке программного обеспечения // Всероссийская конф. с междун. участием "Знания—Онтологии—Теории" (ЗОНТ-07), 14—16 сентября 2007, Новосибирск, Россия. Новосибирск: Институт математики. 2007. В 2 т. Т. 1. С. 122—129.

2. **Jasper R. and Uschold M.** A Framework for Understanding and Classifying Ontology Applications. URL: http://www.informatik.unitrier.de/~ley/db/indices/a-tree/u/Uschold:Michael.html.

3. **Staab S. and Maedche A.** Knowledge Portals: Ontologies at Work // In AI Magazine. 2001. 22 (2). P. 63—75.

4. **Zhdanova A. V.** The People's Portal: Ontology Management on Community Portals. URL: http://www.ee.surrey.ac.uk/Personal/A. Zhdanova/publications.htm.

5. **Грибова В. В.** Расширяемый инструментарий для разработки пользовательского интерфейса, основанный на методах искусственного интеллекта // X Национальная конф. по искусственному интеллекту с междунар. участием. Тр. конф. В 3-т. Т. 1. М.: Физмалит. 2006. С. 125—132.

6. **Denny M.** Ontology Building: a Survey of Editing Tools // http://www.xml.com/pub/a/2004/07/14/onto.html.

7. **Клещев А. С., Орлов В. А.** Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации // Информационные технологии. 2006. № 5. С. 25—31.

8. http://www.alphaworks.ibm.com/contentnr/semanticsfaqs.

9. http://www.w3.org/TR/webont-req/.

10. **Corcho O., Fernandez-Lopez M., Gomez-Perez A.** Methodologies, tools and languages for building ontologies. Where is their meeting point? // Data & Knowledge Engineering. 2003. № 46. P. 41—64.

11. **Cristani M., Cuel R.** A Survey on Ontology Creation Methodologies // In Int. J. on Semantic Web & Information Systems. 2005. 1 (2). P. 48—68.

12. **Jones D., Bench-Capon T. and Visser P.** Methodologies For Ontology Development. URL: http://www.iet.com/Projects/RXF/SME/methodologies-for-ontology-development.pdf.

13. **Noy N., McGuinness D. L.** Ontology Development 101: A Guide to Creating Your First Ontology // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. URL:

http://protege.stanford.edu/publications/ontology_development/ontology101. html.

14. **Guarino N., Carrata M., Giaretta P.** An ontology of meta-level categories // In the Proceeding of the Fourth Int. conf. "Principles of Knowledge representation and Reasoning". Morgan Kaufman, San Mateo, CA. 1994. P. 270—280.

15. **Клещев А. С., Артемьева И. Л.** Отношения между онтологиями предметных областей. Ч. 2. Отношения сходства онтологий, композиция онтологий // Научно-техн. информация. Сер. 2. 2002. № 2. С. 24—31.

16. **Клещев А. С., Москаленко Ф. М., Черняховская М. Ю.** Модель онтологии предметной области "Медицинская диагностика". В 2-х частях. // Научно-техническая информация. Сер. 2. Ч. 1. 2005. № 12. С. 1—7. Ч. 2. 2006. № 2. С. 19—30.

17. **Артемьева И. Л., Рештаненко Н. В., Цветников В. А.** Многоуровневая онтология химии // Всероссийская конф. с междун. участием "Знания—Онтологии—Теории". Новосибирск, 14—16 сентября 2007. Новосибирск: Институт математики. Т. 1. 2007. С. 138—146.

18. **Артемьева И. Л., Цветников В. А.** Фрагмент онтологии физической химии и его модель // Исследовано в России [Электронный ресурс]: многопредметн. научн. журн. Долгопрудный: МФТИ. 2002. № 5. С. 454—474. http://zhurnal.ape.relarn.ru/articles/2002/042.pdf.

19. **Артемьева И. Л., Высоцкий В. И., Рештаненко Н. В.** Описание структурного строения органических соединений в модели онтологий органической химии // Научно-техн. инф. Сер. 2. 2006. № 2. С. 11—19.

20. **Артемьева И. Л., Высоцкий В. И., Рештаненко Н. В.** Модель онтологии предметной области (на примере органической химии) // Научно-техн. инф. Сер. 2. 2005. № 8. С. 19—27.

21. **Артемьева И. Л., Мирошниченко Н. Л.** Модель онтологии рентгенофлуоресцентного анализа // Информатика и системы управления. 2005. № 2. С. 78—88.

22. **Артемьева И. Л., Князева М. А., Купневич О. А.** Модель онтологии предметной области "Оптимизация последовательных программ". В 3-х частях // Научно-техн. инф. Сер. 2. Ч. 1. 2002. № 12. С. 23—28; Ч. 2. 2003. № 1. С. 22—29; Ч. 3. 2003. № 2. С. 27—34.

23. **Клещев А. С., Артемьева И. Л.** Математические модели онтологий предметных областей. Ч. 2. Компоненты модели // Научно-техн. инф. Сер. 2. 2001. № 3. С. 19—29.

24. **Artemieva I. L.** Multilevel ontologies for domains with complicated structures // In the Proceedings of the XШ-th International Conference "Knowledge-Dialog-Solution" — KDS 2007, June 18—24, Varna, Bulgaria, Sofia: FOI ITHEA. 2007. Vol. 2. P. 403—410.

УДК 004.934

А. Л. Ронжин, канд. техн. наук, доц., зав. лаб.,
Санкт-Петербургский институт информатики
и автоматизации РАН

Сравнительный анализ и оценка моделей словаря для систем распознавания русской речи

Проводится анализ трех моделей представления словаря распознавания: линейной модели, лексического дерева и двухуровневого морфофонемного префиксного графа (ДМПГ). Представление словаря в виде списка слов и их транскрипций, которое используется в большинстве современных систем распознавания речи и достаточно успешно подходит для английского, не годится для флективных языков по скорости обработки вследствие их богатой морфологии. Декомпозиция транскрипции каждой словоформы из словаря на основу и окончание с последующим объединением одинаковых последовательностей первых фонем основ и одинаковых транскрипций окончаний обеспечивает формирование компактной морфофонетической структуры словаря в виде ДМПГ. Сложность топологий различных способов представления словаря оценивается по числу узлов и дуг, а также плотности графа словаря. Проверка моделей осуществлялась на словаре, содержащем свыше 2 млн словоформ. Также проанализировано, как изменяются параметры моделей в зависимости от размера словаря.

Ключевые слова: *автоматическое распознавание речи; лексическое дерево; префиксный граф; флективные языки; сверхбольшой словарь.*

Введение

Автоматическое распознавание естественной речи предполагает работу со сверхбольшими словарями, размер которых превышает несколько миллионов словоформ, поэтому разработка средств компактного хранения, скоростного поиска и своевременного отсечения маловероятных гипотез в процессе декодирования является крайне актуальной задачей, особенно для русского языка с относительно высоким уровнем флективности. Для компактного представления словаря транскрипций флективных языков признано эффективным разложение словоформы на сублексические единицы, так как это позволяет сократить размер словаря системы распознавания и, соответственно, повысить скорость декодирования речевого сигнала [1]. Разложение на основе статистических моделей позволяет сильнее сократить размер словаря, но увеличивает риск возникновения грамматически некорректных последовательностей сублексических единиц, которые, тем не менее, с акустической точки зрения являются наиболее правдоподобными [2].

Классической моделью словаря (слов или морфов) является структура, представляющая собой список всех словоформ и их транскрипций (рис. 1, *а*). Транскрипция каждого слова представляет собой цепочку составляющих ее фонем. Модель фонемы обычно строится на основе скрытых моделей Маркова (СММ) и лево-правой модели Бэкиса. Более точное распознавание фонем достигается путем учета фонетического контекста и построения моделей трифонов, а также

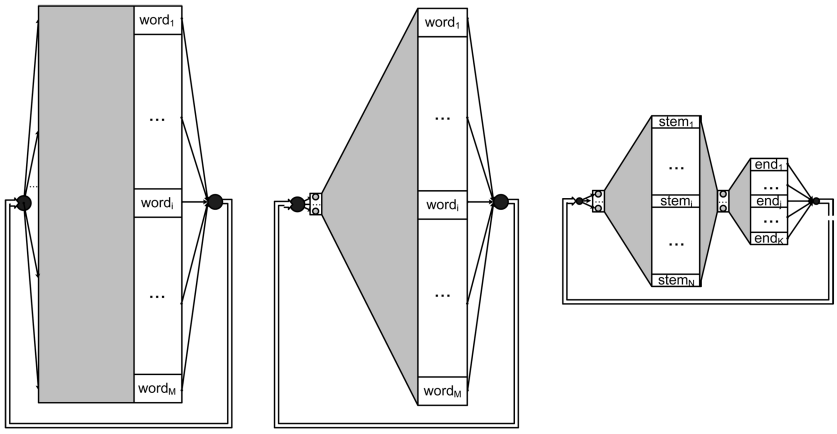


Рис. 1. Способы представления словаря:
а — линейная модель; *б* — лексическое дерево; *в* — ДМПГ

применения смесей гауссовских плотностей распределения вероятностей векторов наблюдений в состояниях фонем.

С помощью СММ обеспечивается объединение моделей фонем, слов, фраз в единую структуру графа, обеспечивающего поиск лучшей гипотезы распознавания. При проектировании системы распознавания речи в зависимости от размера словаря и типа модели языка, которая используется при построении моделей фраз, меняется в основном структура (*lattice*) графа. Поэтому методы параметрического представления речи, методы оценки вероятности состояний, фонем, фраз остаются практически неизменными, а проводится наполнение и оптимизация графа словаря декодера.

С увеличением размера словаря появляются слова с одинаковыми начальными участками, соответственно их транскрипции будут иметь одинаковые начальные фонемы. Объединяя начальные участки транскрипций, словарь преобразуется в лексикофонетическое дерево (рис. 1, *б*), за счет чего достигается значительное сокращение памяти [3]. Прохождение по дереву позволяет синтезировать все возможные слова из словаря. Существующие методы распознавания на основе префиксного лексикофонетического дерева успешно применяются для английского и других языков [4, 5].

Для компактного представления словаря транскрипций в данной работе предлагается использовать декомпозирование словоформы на основу и концовку с помощью морфоанализатора [6], построенного на базе правил словообразования и словоизменения, что позволяет хранить словарь в виде префиксного дерева основ и автоматически генерировать произвольную словоформу [7].

Полученное лексическое префиксное дерево имеет двухуровневую структуру, где первый уровень представляет собой граф основ, а второй уровень — список концовок (элементы, следующие за основой, могут состоять из словообразовательных и словоизменительных суффиксов, окончания и постфикса). Данный двухуровневый морфофонемный префиксный граф (ДМПГ) наиболее компактно описывает все используемые словоформы и их транскрипции (рис. 1, *в*). Генерация ДМПГ проводится по списку транскрибированных словоформ ПО, и поэтому полученный граф способен генерировать только грамматически правильные слова. Для использования данного графа в задаче распознавания слитной речи вводится обратная связь, обеспечивающая генерацию последовательности словоформ с неограниченной длиной. Строго говоря, число слов в последовательности будет зависеть от длины записанного речевого сигнала и при поступлении последней фонемы гипотеза распознанной фразы (путь по графу) заканчивается последним начатым словом.

Для оценки предложенного способа представления словаря проведем сравнительный анализ ДМПГ с двумя общепринятыми моделями представления словаря: моделью списка всех словоформ и лексическим деревом. Прежде всего кратко рассмотрим структуры каждого из трех способов представления словаря. На рис. 1, *а* представлен наиболее простой способ хранения словоформ в виде списка, в каждой строке которого содержится слово и его транскрипция. Более компактное представление достигается путем пофонемного объединения идентичных начальных участков слов и построения лексического дерева (рис. 1, *б*). Наконец, разработанный способ представления словаря на основе ДМПГ схематично представлен на рис. 1, *в*.

Число узлов и дуг, а также плотность графа словаря используются для оценки сложности топологии различных способов представления [8]. Отдельно приведем статистику по узлам разного типа (узлы фонем, словоформ, основ, концовок). Плотность графа вычисляется как отношение суммарного числа всех узлов и дуг к числу словоформ, которые хранятся в данной модели словаря. Плотность графа позволяет оценить среднее число узлов и дуг, которое требуется для представления отдельной словоформы. Далее приведем формулы расчета перечисленных параметров для всех трех способов.

Расчет параметров линейной модели словаря

Размер словаря при использовании линейной модели списка слов пропорционален произведению числа всех словоформ и средней длины сло-

ва. При этом модель каждого слова представляет собой цепочку узлов фонем для описания транскрипции и узел словоформы. Формула для расчета числа узлов фонем $N_{\text{phon\_node}}(List)$ в модели представления словаря в виде списка выглядит следующим образом:

$$N_{\text{phon\_node}}(List) = \sum_{i=1}^{N_{\text{word}}} l_i, \qquad (1)$$

где N_{word} — число уникальных словоформ в словаре; l — число фонем в транскрипции отдельной словоформы. Число узлов $N_{\text{word\_node}}(List)$, содержащих словоформы, будет равно числу словоформ:

$$N_{\text{word\_node}}(List) = N_{\text{word}}. \qquad (2)$$

При объединении всех моделей в единый граф добавляется еще один начальный узел, поэтому общее число узлов $N_{\text{node}}(List)$ в графе будет равно

$$N_{\text{node}}(List) = 1 + N_{\text{phon\_node}}(List) +$$

$$+ N_{\text{word\_node}}(List) = 1 + \sum_{i=1}^{N_{\text{word}}} l_i + N_{\text{word}}. \qquad (3)$$

Так как из начального узла исходит дуга к первой фонеме каждого слова, а из всех узлов словоформ установлена обратная связь в начальный узел, то число дуг в графе равно

$$N_{\text{arc}}(List) = N_{\text{phon\_node}}(List) +$$

$$+ 2N_{\text{word\_node}}(List) = \sum_{i=1}^{N_{\text{word}}} l_i + 2N_{\text{word}}. \qquad (4)$$

Тогда суммарное число узлов и дуг $N_{\text{node \& arc}}(List)$ в графе

$$N_{\text{node \& arc}}(List) = N_{\text{node}}(List) + N_{\text{arc}}(List) =$$

$$= 1 + 2\sum_{i=1}^{N_{\text{word}}} l_i + 3N_{\text{word}}. \qquad (5)$$

Плотность графа $N_{\text{density}}(List)$ рассчитываем как отношение суммарного числа узлов и дуг к числу слов в словаре, используя следующую формулу:

$$N_{\text{density}}(List) = \frac{N_{\text{node \& arc}}(List)}{N_{\text{word}}} =$$

$$= \frac{1 + 2\sum_{i=1}^{N_{\text{word}}} l_i + 3N_{\text{word}}}{N_{\text{word}}}. \qquad (6)$$

Расчет параметров модели лексического дерева

Теперь рассмотрим формулы для вычисления параметров другого способа представления словаря на базе лексического дерева. Посредством префиксного лексического дерева достигается значительное сокращение элементов графа за счет объединения узлов фонем на начальных участках одинаковых фонетических транскрипций. Так как число узлов фонем $N_{\text{phon\_node}}(Tree)$ сокращается в процессе построения дерева и зависит от подобия префиксов фонетических транскрипций слов предметной области, то его аналитическую формулу вывести невозможно, а расчет $N_{\text{phon\_node}}(Tree)$ проводится путем простого пересчета узлов фонем после построения дерева.

Тем не менее, чтобы лучше представлять, как происходит процесс ветвления в дереве, рассмотрим, как изменяется число узлов, стоящих на равноудаленных от начального узла позициях. Другими словами, необходимо проанализировать, сколько узлов используется на различных срезах дерева. Из начального узла выходят дуги к узлам уникальных фонем, с которых начинаются все словоформы. С возникновением различий в транскрипциях происходит ветвление, и повторяющиеся фонемы появляются на одном срезе в различных фонетических путях. Число фонем в транскрипциях словоформ или длина фонетических путей изменяется от 1 до длины транскрипции самого длинного слова $l_{\max}^{\text{word}}$, тогда общее число узлов фонем $N_{\text{phon\_node}}(Tree)$ можно рассчитать, как сумму узлов на каждом срезе $slice_i^{\text{tree}}$ в дереве:

$$N_{\text{phone\_node}}(Tree) = \sum_{i=1}^{l_{\max}^{\text{word}}} N_{\text{phon}}(slice_i^{\text{tree}}). \qquad (7)$$

При движении от начального узла вглубь дерева вначале число узлов фонем на каждом срезе увеличивается, а после прохождения среза, равного средней длине транскрипции из данного словаря, их число начинает сокращаться, поскольку часть фонетических путей заканчивается узлами словоформ. Распределение числа узлов по срезам дерева близко к нормальному. Следует заметить, что в модели списка число узлов фонем на первом срезе равно числу словоформ N_{word} в словаре и затем постепенно убывает по мере удаления от начального узла, а их суммарное число может быть рассчитано по аналогичной формуле:

$$N_{\text{phon\_node}}(List) = \sum_{i=1}^{l_{\max}^{\text{word}}} N_{\text{phon}}(slice_i^{\text{list}}). \qquad (8)$$

Число узлов $N_{\text{word\_node}}(Tree)$, содержащих словоформы, остается таким же, как и в базовом способе на основе списка моделей, и равно числу словоформ в словаре:

$$N_{\text{word\_node}}(Tree) = N_{\text{word}}. \qquad (9)$$

Начальный узел также необходим в лексическом дереве, поэтому общее число узлов $N_{\text{node}}(Tree)$ будет равно:

$$N_{\text{node}}(Tree) = 1 + N_{\text{phon\_node}}(Tree) + N_{\text{word\_node}}(Tree) = 1 + \sum_{i=1}^{l_{\max}^{\text{word}}} N_{\text{phon}}(slice_i^{\text{tree}}) + N_{\text{word}}. \quad (10)$$

Число дуг в дереве рассчитывается аналогично, как в модели списка, однако в данном случае число узлов фонем вычисляется по своей формуле:

$$N_{\text{arc}}(List) = N_{\text{phon\_node}}(Tree) + 2N_{\text{word\_node}}(Tree) = \sum_{i=1}^{l_{\max}^{\text{word}}} N_{\text{phon}}(slice_i^{\text{tree}}) + 2N_{\text{word}}. \quad (11)$$

Суммарное число узлов и дуг $N_{\text{node \& arc}}(Tree)$ в лексическом дереве:

$$N_{\text{node \& arc}}(Tree) = N_{\text{node}}(Tree) + N_{\text{arc}}(Tree) = 1 + 2\sum_{i=1}^{l_{\max}^{\text{word}}} N_{\text{phon}}(slice_i^{\text{tree}}) + 3N_{\text{word}}. \quad (12)$$

Плотность графа для дерева $N_{\text{density}}(Tree)$ рассчитывается по следующей формуле:

$$N_{\text{density}}(Tree) = \frac{N_{\text{node \& arc}}(Tree)}{N_{\text{word}}} = \frac{1 + 2\sum_{i=1}^{l_{\max}^{\text{word}}} N_{\text{phon}}(slice_i^{\text{tree}}) + 3N_{\text{word}}}{N_{\text{word}}}. \quad (13)$$

Расчет параметров ДМПГ

С помощью модели дерева достигается значительное сокращение узлов фонем в графе. В то же время ДМПГ, построенный по принципам префиксного лексического дерева, сохраняет его преимущества и имеет двухуровневую морфологическую структуру. За счет этого сложность ДМПГ пропорциональна числу основ в словаре. Для оценки сложности топологии ДМПГ далее рассмотрим аналогичные формулы определения числа узлов и дуг, использованных при построении графа. Так как граф имеет двухуровневую структуру, то для каждого параметра расчет будет проводиться в три этапа: анализ уровня основ, анализ уровня концовок и суммарная оценка.

Так как первый уровень ДМПГ представляет собой лексическое дерево основ, то расчет числа узлов фонем на первом уровне $N_{\text{phon\_node\_stem}}$(ДМПГ) проводится путем суммирования узлов на каждом срезе $slice_i^{\text{ДМПГ}}$. При этом число фонем в транскрипциях словоформ или длина фонетических путей изменяется от 1 до числа фонем в транскрипции самой длинной основы $l_{\max}^{\text{stem}}$, а не словоформы:

$$N_{\text{phon\_node\_stem}}(\text{ДМПГ}) = \sum_{i=1}^{l_{\max}^{\text{stem}}} N_{\text{phon}}(slice_i^{\text{ДМПГ}}). \quad (14)$$

Учитывая, что транскрипции концовок располагаются в графе независимо друг от друга, расчет числа узлов фонем $N_{\text{phon\_node\_ending}}$(ДМПГ) на втором уровне проводится следующим образом:

$$N_{\text{phon\_node\_ending}}(\text{ДМПГ}) = \sum_{i=1}^{N_{\text{ending}}} l_i, \quad (15)$$

где N_{ending} — число концовок в словаре; l_i — число фонем в транскрипции концовки с номером i. Отметим, что в графе хранятся только уникальные концовки и их транскрипции. Суммарное число узлов фонем $N_{\text{phon\_node}}$(ДМПГ), включая уровни основ и концовок, будет равно

$$N_{\text{phom\_node}}(\text{ДМПГ}) = N_{\text{phone\_node\_stem}}(\text{ДМПГ}) + N_{\text{phone\_node\_ending}}(\text{ДМПГ}) = \sum_{i=1}^{l_{\max}^{\text{stem}}} N_{\text{phon}}(slice_i^{\text{ДМПГ}}) + \sum_{i=1}^{N_{\text{ending}}} l_{\text{i}}. \quad (16)$$

Так как в графе существуют узлы только уникальных основ N_{stem} и концовок N_{ending}, то общее число узлов $N_{\text{word\_node}}$(ДМПГ), содержащих полные транскрипций и индексы лексических единиц, равно

$$N_{\text{word\_node}}(\text{ДМПГ}) = N_{\text{stem}} + N_{\text{ending}}. \quad (17)$$

Суммарное же число всех узлов, включая начальный, будет равно

$$N_{\text{node}}(\text{ДМПГ}) = 1 + N_{\text{phon\_node}}(\text{ДМПГ}) + N_{\text{stem}} + N_{\text{ending}} = 1 + \sum_{i=1}^{l_{\max}^{\text{stem}}} N_{\text{phon}}(slice_i^{\text{ДМПГ}}) + \sum_{i=1}^{N_{\text{ending}}} l_i + N_{\text{stem}} + N_{\text{ending}}. \quad (18)$$

Число дуг в графе складывается из нескольких составляющих. Во-первых, это число дуг $N_{\text{phon\_node}}$(ДМПГ) + N_{stem}, задействованных в лексическом дереве основ и списке концовок, затем дуги $N_{\text{arc\_stem\_ending}}$, связывающие основы и концовки, в количестве, необходимом для построения всех возможных словоформ; и, наконец, дуги обратных связей N_{ending}:

$$N_{\text{arc}}(\text{ДМПГ}) = N_{\text{phon\_node}}(\text{ДМПГ}) + N_{\text{stem}} + N_{\text{arc\_stem\_ending}} + N_{\text{ending}} =$$

$$= \sum_{i=1}^{l_{\max}^{\text{stem}}} N_{\text{phon}}(slice_i^{\text{ДМПГ}}) + \sum_{i=1}^{N_{\text{ending}}} l_i + \\ + N_{\text{stem}} + N_{\text{arc\_stem\_ending}} + N_{\text{ending}}. \quad (19)$$

Тогда суммарное число узлов и дуг $N_{\text{node \& arc}}$(ДМПГ) в графе будет равно

$$N_{\text{node \& arc}}(\text{ДМПГ}) = \\ = N_{\text{node}}(\text{ДМПГ}) + N_{\text{arc}}(\text{ДМПГ}) = \\ = 1 + 2\left[\sum_{i=1}^{l_{\max}^{\text{stem}}} N_{\text{phon}}(slice_i^{\text{ДМПГ}}) + \right. \\ \left. + \sum_{i=1}^{N_{\text{ending}}} l_i + N_{\text{stem}} + N_{\text{ending}}\right] + N_{\text{arc\_stem\_ending}}. \quad (20)$$

Плотность графа N_{density}(ДМПГ) рассчитывается по следующей формуле:

$$N_{\text{density}}(\text{ДМПГ}) = \frac{N_{\text{node \& arc}}(\text{ДМПГ})}{N_{\text{word}}} = \\ = \frac{1 + 2\left[\sum_{i=1}^{l_{\max}^{\text{stem}}} N_{\text{phon}}(sclice_i^{\text{ДМПГ}}) + \sum_{i=1}^{N_{\text{ending}}} l_i + N_{\text{stem}} + N_{\text{ending}}\right] + N_{\text{arc\_stem\_ending}}}{N_{\text{word}}}. \quad (21)$$

Сравнительный анализ параметров моделей

В этом разделе с помощью приведенных выше формул постараемся оценить преимущество предложенного способа представления словаря. Во всех трех моделях число хранящихся словоформ N_{word} остается неизменным. За счет объединения идентичных фонем на первых срезах лексического дерева достигается существенное сокращение числа узлов фонем по сравнению с моделью списка, где число узлов фонем на первом срезе равно числу словоформ в словаре, а затем постепенно убывает. В отличие от лексического дерева в ДМПГ строится дерево только для основ, а не для полных словоформ. Кроме того, в виде списка хранятся фонетические пути только для уникальных окончаний, в то время как в лексическом дереве окончания повторяются в парадигмах всех слов из словаря.

Число узлов, содержащих слова в лексическом дереве и в модели списка, является одинаковым, так как в том и в другом случае в узлах хранятся все словоформы из словаря. В графе ДМПГ узлов, содержащих полные словоформы, не существует, так на этапе подготовки словаря осуществляется декомпозиция всех словоформ на основы и концовки.

В структуре лексического дерева проводится сокращение узлов фонем, однако в листе каждой ветви дерева по-прежнему хранится полная словоформа. Поэтому оптимизация способа представления фонетических путей не обеспечивает сокращения узлов словоформ в модели дерева. В графе ДМПГ обеспечивается сокращение как узлов фонем, так и узлов слов, благодаря декомпозиции словоформы на основу и концовку. При этом число уникальных основ и концовок, которые хранятся в узлах графа, будет существенно меньше, чем число уникальных словоформ. Так как число уникальных словоформ N_{word} в каждом из способов одинаковое, то отношение плотностей графов равно отношению суммарного числа узлов и дуг.

В следующем разделе на примере конкретного словаря экспериментально проведем сравнение параметров для всех трех моделей. Особый интерес представляет анализ распределения числа узлов фонем по срезам, поскольку главным образом этот фактор влияет на сложность топологии как графа ДМПГ, так и лексического дерева. Параллельно проведена оценка и других параметров модели списка и лексического дерева с использованием приведенных выше формул. Для экспериментальной проверки разработанного способа используем в качестве базового словаря грамматический словарь А. А. Зализняка [9].

Экспериментальная проверка моделей на сверхбольшом словаре

Для тестирования моделей был сформирован список всех словоформ и их транскрипций путем обработки грамматического словаря А. А. Зализняка. Сравнительный анализ моделей по параметрам, описанным в предыдущих разделах, показал явное преимущество ДМПГ. Характеристики графов, построенные по трем разным подходам, представлены в таблице. ДМПГ, описывая точно

Сравнение ДМПГ с другими моделями представления словаря

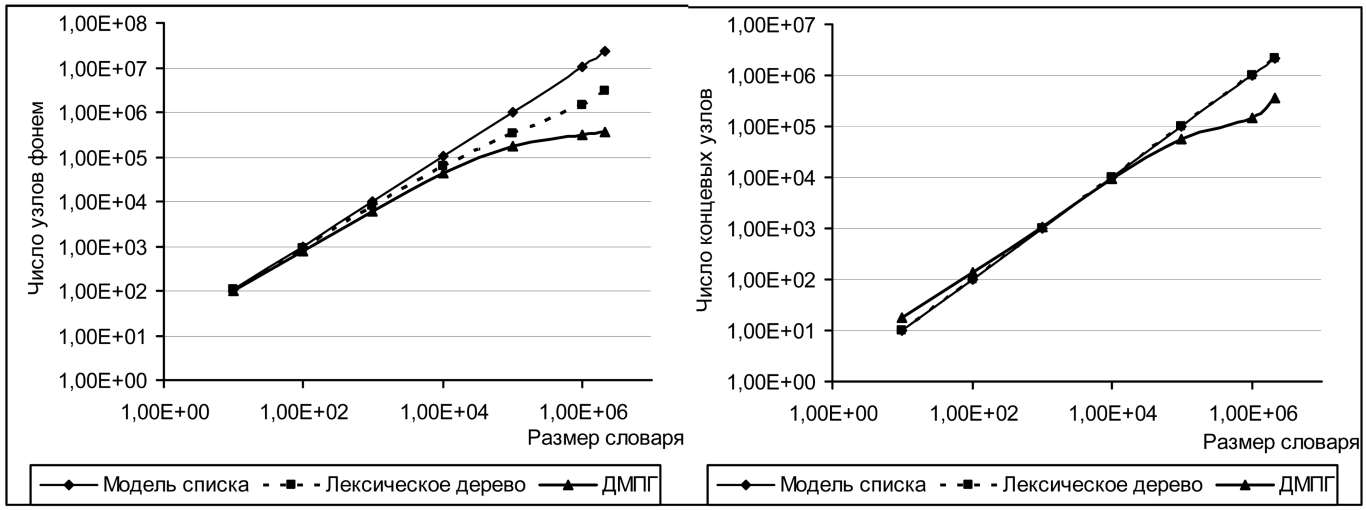
Критерий сравнения	Вид представления словаря		
	Список цепочек фонем	Лексическое дерево	ДМПГ
Число узлов фонем	23 017 898	2 967 752	371 018
Сокращение числа узлов фонем, раз	—	7,75	**62,03**
Число узлов словоформ (основ + концовок)	2 095 659	2 095 659	187 996
Суммарное число узлов	25 113 558	5 063 412	559 015
Число дуг	27 209 216	7 159 070	747 010
Суммарное число узлов и дуг	52 322 774	12 222 482	1 306 025
Плотность графа словаря	24,96	5,83	0,62
Сокращение плотности графа словаря, раз	—	4,28	**40,06**

такой же словарь, как и основные модели, использует в 62 раза меньше узлов фонем, а также имеет в 40 раз меньшую плотность графа.

Также было проанализировано, как изменяются параметры моделей в зависимости от размера словаря (рис. 2). Сокращенные словари создавались путем случайного отбора заданного числа уникальных словоформ из базового словаря. По суммарному числу узлов ДМПГ имеет явное преимущество для словарей, имеющих размер, начиная с 10 000 словоформ. По остальным показателям, в том числе по плотности графа словаря, ДМПГ лидирует уже после размера 100 словоформ.

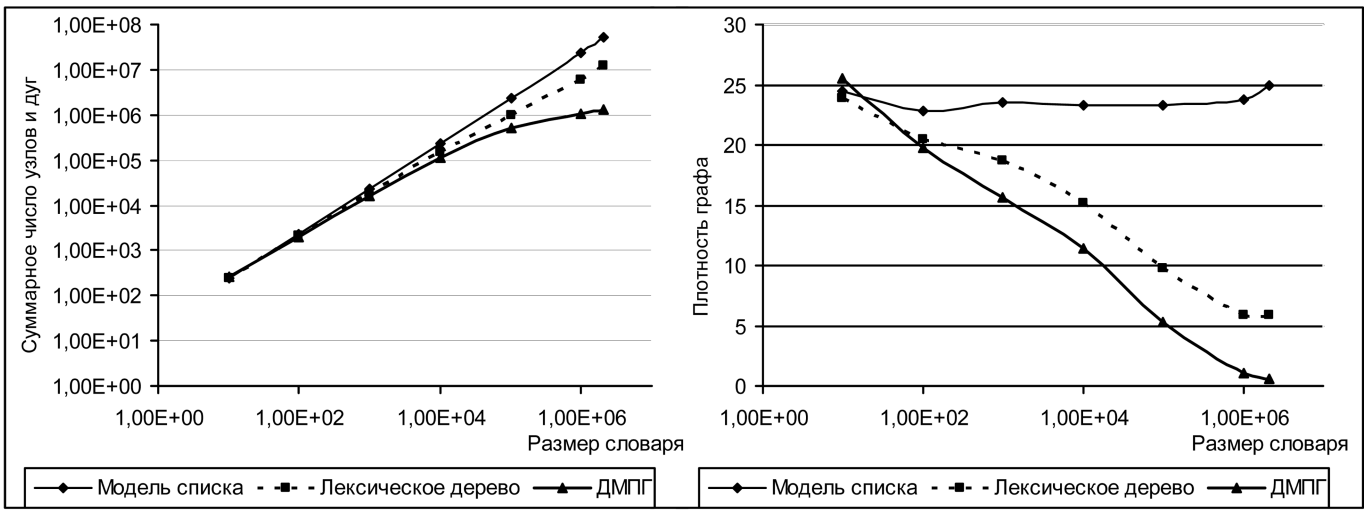
Особый интерес представляет распределение числа узлов фонем по срезам графа (рис. 3), поскольку главным образом этот фактор влияет на сложность топологии как графа ДМПГ, так и лексического дерева. Проанализируем все три модели отдельно.

В линейной модели списка транскрипций словоформ на первом срезе присутствуют первые фонемы всех словоформ, поэтому число узлов фонем равно числу словоформ в словаре. Для тестового словаря число узлов фонем первого среза равно 2 095 659. На втором срезе число узлов фонем становится меньше (2 095 627 узлов), поскольку слова, состоящие из одной буквы, уже не участвуют в этом срезе. По мере увеличения номера среза и, соответственно, длины слова, число узлов фонем на каждом уровне сокращается. На последнем 27 срезе присутствуют 16 узлов фонем для последних фонем из 16 самых длинных транскрипций (например, транскрипции "иНТирнацианаЛиЗИравафшеваСа", "супстанцианаЛиЗИравафшеваСа" и другие формы от этих слов). Для повышения компактности и скорости обработки транскрипций мягкость согласных и ударность гласных выделяется регистром.



а)

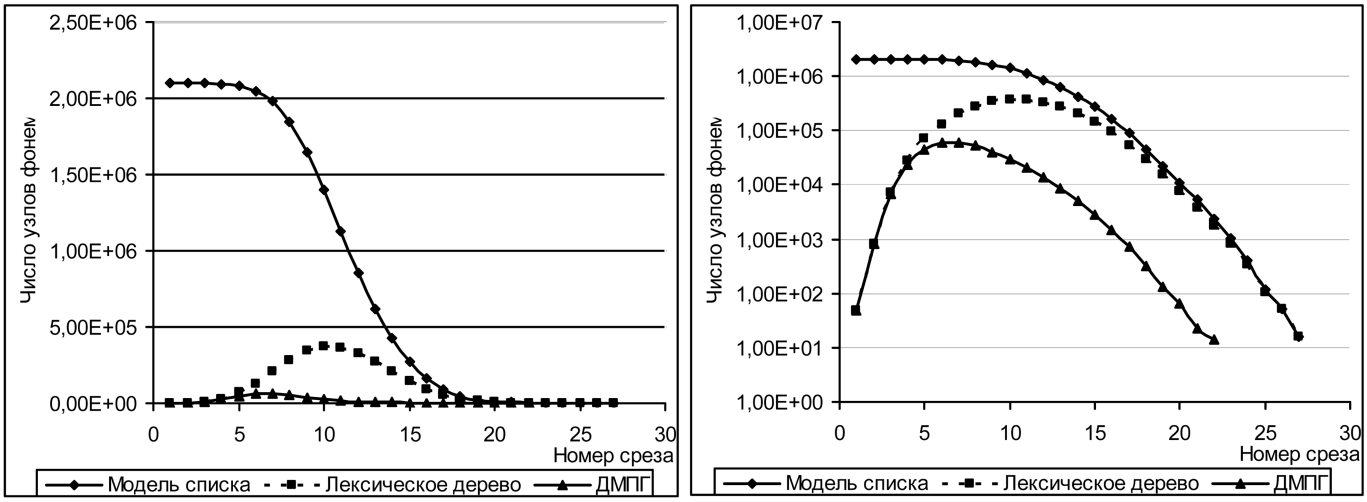
б)



в)

г)

Рис. 2. Сравнение моделей:
а — по числу узлов фонем; *б* — по числу концевых узлов; *в* — по суммарному числу узлов и дуг; *г* — по плотности графа



а) *б*)

Рис. 3. Распределение узлов фонем по срезам моделей:
а — линейная шкала; *б* — логарифмическая шкала

В лексическом дереве на первом срезе присутствуют узлы только уникальных фонем, по мере появления различий в цепочках фонем число узлов на каждом последующем срезе увеличивается. После достижения среза с номером, равным средней длине транскрипции, большая часть транскрипций постепенно заканчивается узлами словоформ, и поэтому число узлов фонем начинает сокращаться. На первом и последнем срезах для данного словаря находится 48 и 18 узлов фонем, соответственно. Максимальное число узлов фонем 371 545 достигается на 10-м срезе.

Для ДМПГ лексическое дерево строится для основ, а не для полных словоформ, и поэтому в среднем длина транскрипций основ будет меньше. Для тестового словаря распределения числа транскрипций по длинам отдельно для основ и словоформ приведены на рис. 4. Наибольшее число транскрипций словоформ состоят из 10 фонем и соответственно на графе описываются в виде цепочки из 10 узлов фонем. Большинство транскрипций основ имеет длину 6 и, следовательно, срез с максимальным числом узлов фонем будет наблюдаться раньше. Так, уже после шестого среза, где содержатся 60 445 узлов фонем, начинается их сокращение и на последнем 22-м срезе присутствуют 14 фонем.

Следует отметить, что самые длинные транскрипции основ, которые получились в ходе формирования ДМПГ для данного словаря (например, "воздухараспРиДиЛИТеЛн", "праТиваправИТеЛстВеНен") не являются основами самых длинных транскрипций словоформ. Это связано с тем, что наиболее длинные транскрипции наблюдаются у причастий, а поскольку при декомпозиции в них выделяется достаточно длинная концовка, то длина транскрипции основы становится значительно меньшей по сравнению с длиной транскрипции словоформы. Например, при декомпозиции одной из самых длинных транскрипций "иНТирнацианаЛиЗИрава/фшеваСа" выделятся основа, содержащая 20 знаков, и концовка — 7 знаков.

Поскольку в ДМПГ хранятся только уникальные концовки, то при формировании транскрипций словоформ по графу алгоритм будет использовать одни и те же концовки несколько раз. Во избежание многократного подсчета узлов одной и той же концовки при анализе запоминаются все использованные концовки и учитываются узлы фонем только концовки, которая встретилась впервые. Например, при обработке транскрипции "иНТирнацианаЛиЗИрава/фшеваСа" подсчет узлов фонем заканчивается уже на 20-м знаке, так как концовка "фшеваСа" была использована ранее в транскрипции "абаНИрава/фшеваСа". В результате декомпозиции словоформы на основу и концовку, а также учета транскрипций только новых концовок число срезов в ДМПГ на пять меньше по сравнению с линейной моделью и лексическим деревом.

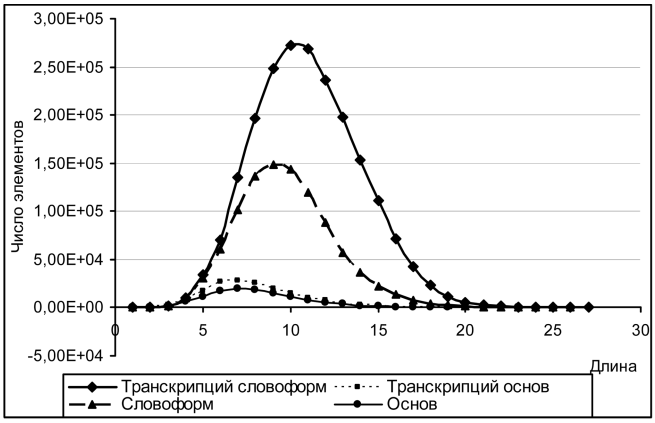


Рис. 4. Распределение длин основ, словоформ и их транскрипций для тестового словаря

Таким образом, для данного словаря, включающего 2 095 659 уникальных транскрипций словоформ, предложенный способ представления словаря на базе ДМПГ показал заметное преимущество. Используя лексическое дерево для представления транскрипций основ и объединяя одинаковые концовки, срез с максимальным числом узлов фонем получаем почти в 2 раза быстрее, а значение максимума — в 6 раз меньше в ДМПГ, чем в лексическом дереве.

Заключение

Разработка компактного способа представления словаря особенно актуальна для флективных языков с богатой морфологией. Предложенная модель двухуровневого морфофонетического префиксного графа за счет декомпозиции словоформ на основу и окончание по грамматическим правилам позволяет компактно хранить словарь в виде префиксного дерева основ и автоматически генерировать произвольную словоформу. Процедура его построения сводится к транскрибированию всех словоформ словаря и последующему объединению начальных участков основ и окончаний в двухуровневый ориентированный граф. Аналитический расчет параметров линейной модели, лексического дерева и ДМПГ показал преимущество предложенного способа. Экспериментальная проверка и сравнение параметров для всех трех моделей проведена на словаре, размером свыше 2 млн слов. Преимущество ДМПГ наблюдается уже на словарях размером более 1000 словоформ. Таким образом, разработанная модель компактного представления словаря обеспечивает значительное сокращение размера памяти и высокую скорость обработки, что важно при создании систем декодирования слитной русской речи со сверхбольшим словарем.

Работа выполняется при поддержке гранта РФФИ № 07-07-00073-а.

Список литературы

1. **Kurimo M., Creutz M., Varjokallio M., Arisoy E., Saraclar M.** Unsupervised segmentation of words into morphemes — Morpho challenge 2005 application to automatic speech recognition // Proc. Interspeech 2006. Pittsburgh, USA, 2006. P. 1021—1024.

2. **Kneissler J., Klakow D.** Speech recognition for huge vocabularies by using optimized subword units // Proc. Eurospeech 2001. Aalborg, Denmark, 2001. P. 69—72.

3. **Ortmanns S., Eiden A., Ney H.** Improved Lexical Tree Search for Large Vocabulary Recognition // IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Seattle, WA, 1998. P. 817—820.

4. **Demuynck K., Duchateau J., Van Compernolle P., Wambacq P.** An efficient search space representation for large vocabulary continuous speech recognition // Speech Communication. 2000. Vol. 30. N 1. P. 37—53.

5. **Pražák A., Psutka J., Hoidekr J., Kanis J., Müller L., Psutka J.** Adaptive language model in automatic online subtitling // Proc. 2nd IASTED International Conference on Computational Intelligence CI 2006. San Francisco, California, USA, 2006. P. 479—483.

6. **Леонтьева Ан. Б.** Модуль морфофонетической обработки слов для построения словаря распознавателя русской слитной речи // Искусственный интеллект. 2007. № 3. С. 319—327.

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

8. **Hey H., Ortmanns S., Lindam I.** Extensions to the Word Graph Method for Large Vocabulary Continuous Speech Recognition // Proc. of ICASSP'97. 1997. Vol. 3. P. 1787—1790.

9. **Зализняк А. А.** Грамматический словарь русского языка. М.: Русские словари, 2003. 800 с.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УДК 621.3.019.3(004.3.06)

С. Г. Мосин, канд. техн. наук, доц.,
Владимирский государственный университет

Современные тенденции и технологии проектирования интегральных схем

Рассмотрены состояние и прогноз развития мирового рынка микроэлектроники. Представлены современные тенденции изменения интегральной технологии. Описаны технологии проектирования специализированных интегральных схем.

Ключевые слова: *специализированные ИС, интегральные технологии (ИТ), анализ и прогноз развития ИТ.*

Электронная промышленность является одной из самых динамично развивающихся отраслей мирового рынка. Средний ежегодный рост в данной области с 2004 г. составляет порядка 12,6 %. Анализ и прогноз объема мирового товарооборота электронной продукции по основным категориям представлены в таблице. По прогнозам к 2012 г. глобальный рынок микроэлектронных изделий (МЭИ) достигнет примерно 3,2 трлн долл., что составит на ближайшие пять лет (2007—2012 гг.) около 9,5 % среднегодового темпа роста [1].

Основные мировые центры микроэлектронной промышленности расположены в Америке, Европе и Азии. На рис. 1 в процентном соотношении

Мировой объем товарооборота электронной продукции, млрд долл.

Категория	Год					Среднегодовой темп роста (2007—2012 гг.), %
	2004	2005	2006	2007	2012	
Материалы для "кремниевых фабрик"	16,8	18,1	19,4	20,5	27,6	6,1
Полупроводниковые приборы и ИС	213,0	227,5	247,7	264,1	375,1	7,3
Коммуникация	141,0	158,5	183,1	198,6	310,3	9,3
Компьютеры	235,6	295,1	368,9	419,0	711,0	11,2
Бытовая электроника	135,1	183,2	246,0	291,1	641,6	17,1
Промышленная электроника	628,4	665,3	728,4	768,8	1038,0	6,2
Автотранспортная электроника	41,2	44,0	47,1	50,3	67,6	6,1
Всего	1411,1	1591,7	1840,5	2012,4	3171,2	9,5

представлено распределение объемов мирового рынка производства и продаж микроэлектронных изделий в период с 1991 г. по 2007 г. в различных регионах мира. Микроэлектронные компании, расположенные в Америке, обеспечивают примерно половину мирового рынка МЭИ. На фоне этого наблюдается устойчивая тенденция роста объемов производства МЭИ в Азиатско-Тихоокеанском регионе, в первую очередь, за счет активного развития микроэлектронной промышленности в Китае, Таиланде и Южной Корее [2].

В Европе на фоне постепенного снижения общих объемов наблюдается постепенное увеличение числа компаний, занимающихся проектированием МЭИ, но не обладающих собственными производственными мощностями — заводами по изготовлению микросхем (*fabless*), в среднем на 4 % в год. По оценке на конец 2007 г. наибольшее число таких компаний расположено в Великобритании (~ 31,9 %). Общая информация о географическом расположении в Европе fabless-компаний представлена на рис. 2 [3, 4].

Зарождение и развитие микроэлектроники в первую очередь связаны с повышением степени интеграции ИС, которая отражает рост сложности полупроводниковой технологии и проектируемых систем. На сложность полупроводниковой технологии влияют масштабирование технологического процесса, использование новых материалов, а также реализация передовых архитектур устройств и механизмов межсоединений.

В качестве основных параметров, характеризующих интегральную технологию, используют минимальный технологический размер и длину затвора транзистора. Первый параметр отражает специфику плотности размещения компонентов и межсоединений на кристалле ИС. Минимальный технологический размер рассматривают как минимальное расстояние между соседними контактами в первом слое металлизации ИС (*half-pitch metal 1*). Второй параметр определяет геометрические особенности транзистора и его инерционные свойства.

Оценка динамики изменения интегральной технологии за последние десять лет позволяет установить тенденцию масштабирования минимального технологического размера. До 1998 г. цикл перехода от используемой интегральной технологии к следующему поколению составлял три года с масштабирующим множителем примерно 0,71. С 1998 г. технологический цикл составляет два года с сохранением значения масштабирующего множителя (0,71x/2 года). Таким образом, в течение двух последовательных технологических циклов минимальный технологический размер сокращается в 2 раза [5].

Изменение длины затвора при переходе на новую технологию в период до 1999 г. происходило с масштабирующим множителем 0,71 при продолжительности цикла два года (0,71x/2 года). В 1999 г. длина затвора транзистора составила ме-

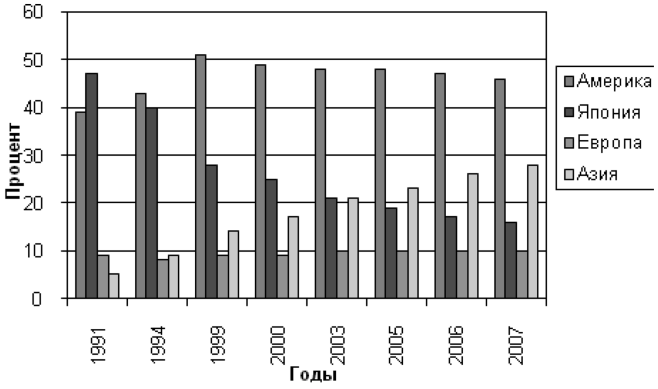


Рис. 1. Соотношение мировых объемов производства МЭИ

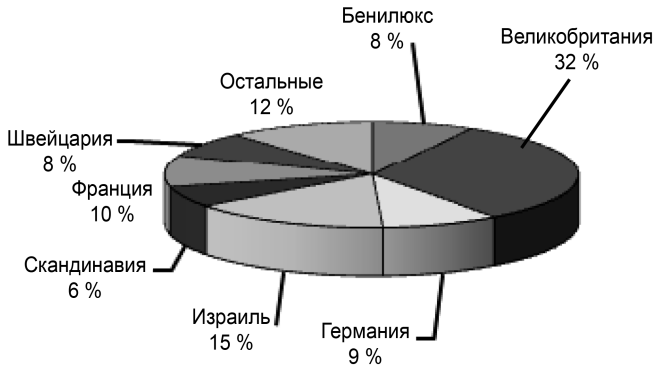


Рис. 2. Процентное соотношение fabless-компаний в странах Европы

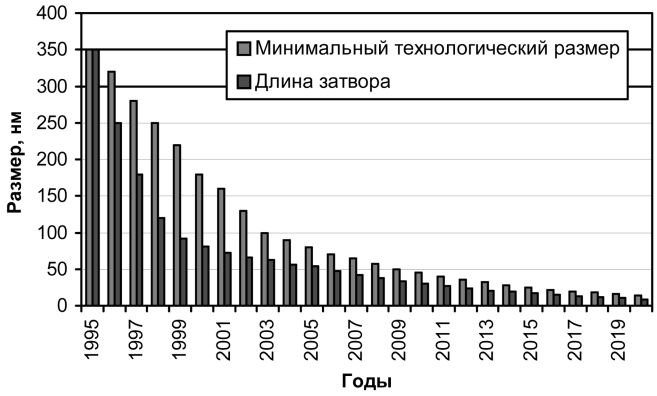


Рис. 3. Динамика изменения технологических норм изготовления ИС

нее 100 нм. Данное достижение позволило неофициально объявить о начале эры нанотехнологий. С этого года продолжительность технологического цикла увеличилась до трех лет с сохранением масштабирующего множителя — 0,71x/3 года.

По аналитическим прогнозам выявленная тенденция для минимального технологического размера и длины затвора транзистора в ближайшей перспективе сохранится (рис. 3) [5].

Снижение длины затвора транзистора существенным образом влияет на изменение задержки распространения сигнала в кристалле ИС. Так, в микросхемах с технологическими размерами до 0,35 мкм до 60 % общей задержки определяла задержка на логических элементах и 40 % — на линиях межсоединений компонентов. С минимизацией длины затвора данное соотношение изменилось существенным образом — общая задержка распространения сигнала в ИС в большей степени зависит от задержек на линиях межсоединений (до 80 %), чем на внутренних логических элементах (20 %). Данное обстоятельство накладывает дополнительные требования к качеству выполнения операций размещения и трассировки элементов в кристалле ИС.

Сложность проектируемых систем определяют числом реализуемых в ИС транзисторов и числом используемых слоев межсоединений (слоев металлизации). Последний параметр непосредственно связан с применяемым технологическим процессом. Минимизация технологических размеров позволила увеличивать число внутренних слоев металлизации в топологии кристалла (рис. 4) [6].

Сложность ИС, а следовательно, и число транзисторов зависят от назначения (класса) проектируемого устройства. На фоне уменьшения размеров транзистора необходимо отметить постоянный рост плотности размещения элементов на единицу площади кристалла. По оценке ITRS [5], при проектировании специализированных заказных ИС (ASIC — *Application Specific Integrated Circuit*) общая сложность устройств возрастает ежегодно в среднем на 26 %. При сохранении такой тенденции к 2012 г. кристалл заказной микросхемы будет содержать около 10 млрд транзисторов при плотности более 1 млрд транзисторов на одном квадратном сантиметре (рис. 5, *а*).

При проектировании процессоров прогнозируют сохранение общей закономерности, высказанной Гордоном Муром (*Moore's law*) еще в 1965 г., в виде удвоения числа транзисторов в микропроцессоре каждые полтора—два года. Плотность размещения транзисторов на единице площади кристалла микропроцессора ежегодно растет, как и у ASIC, на 26 % (рис. 5, *б*).

Анализ рынка микроэлектроники в странах Азиатско-Тихоокеанского региона, наиболее активно развивающегося в данном направлении, показывает устойчивую тенденцию увеличения объемов ASIC-проектов на фоне остальных технологий проектирования электронных устройств [7—9].

Традиционно существенную долю среди специализированных ИС составляют устройства массового потребления низкой (до 250 тыс. транзисторов) и средней сложности (до 2,5 млн транзисторов). К числу устройств низкой сложности относятся микросхемы, используемые в бытовой технике, периферийных устройствах вычислительной техники, электронных открытках и игрушках. Устройства средней сложности реализуют микроконтроллеры различного назначения, элементы теле- и радиотехники, сетевое и мультимедиа оборудование и др. В настоящее время наблюдается увеличение объемов производства специализированных ИС высокой сложности (рис. 6). Хотя общий объем таких микросхем достаточно мал, но ежегодный рост выпуска приложений высокой сложности носит показательный характер.

Другой особенностью ASIC-проектов является постепенный переход от низкочастотных реализа-

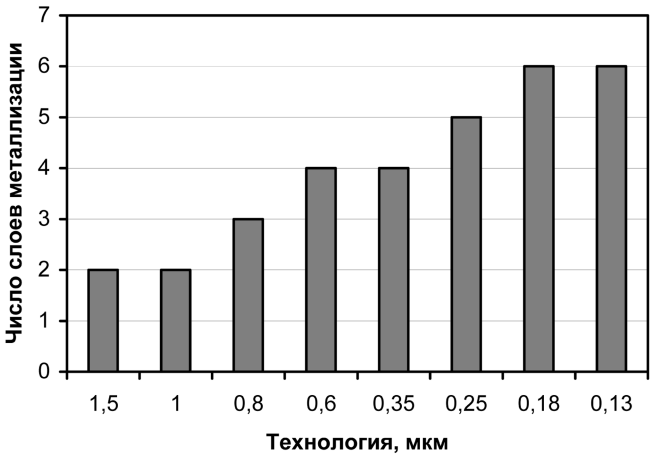


Рис. 4. Число слоев металлизации, реализуемых в интегральных технологиях

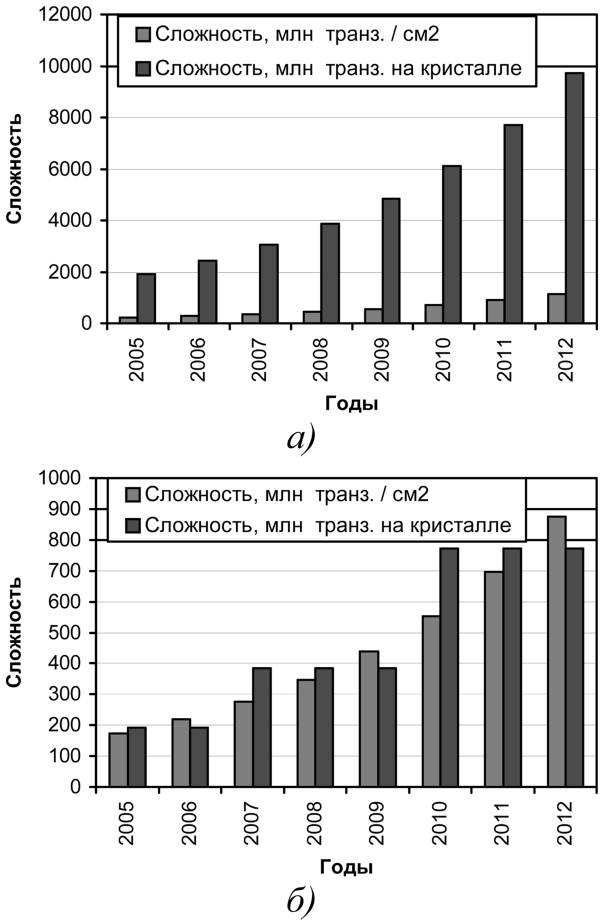


Рис. 5. Динамика роста сложности проекта:
а — для ASIC; *б* — для микропроцессоров

ций (менее 50 МГц) к средне- и высокочастотным (рис. 7). В настоящее время около 50 % специализированных микросхем реализуют в диапазоне тактовых частот 100—250 МГц и более 20 % — в диапазоне выше 250 МГц. Для ASIC-проектов можно добиться максимальных значений тактовых частот, что обеспечивает наилучший показатель быстродействия по сравнению с другими технологиями проектирования.

Во многом указанные достижения стали возможны благодаря оперативному переходу на передовые технологические нормы производства ИС. Проектирование с использованием новых технологий сопряжено со значительными материальными затратами по сравнению с применением предыдущих технологий. Поэтому при разработке устройства важно оценить технико-экономические показатели использования различных технологий и окончательно остановиться на наиболее подходящей для конкретной реализации. Однако разрабатывать устройства с принципиально новыми параметрами и характеристиками, такими как быстродействие, энергопотребление, занимаемая площадь кристалла, сложность и др., можно только с использованием современных интегральных технологий. Показатели процентного соотношения использования компаниями стран Азиатско-Тихоокеанского региона технологических процессов при проектировании специализированных ИС приведены на рис. 8. Данный анализ показывает, что порядка 50 % всех ASIC-проектов в настоящее время реализуют по технологии 130 нм и ниже [7, 8].

Рост объемов реализации ASIC-проектов определяется не только доступностью новых технологических процессов, но и развитием средств автоматизированного проектирования, разработкой эффективных маршрутов проектирования, а также созданием большого числа библиотек решений многократного использования (IP-ядер или блоков). В настоящее время более 90 % компаний, разрабатывающих специализированные ИС, применяют при проектировании IP-ядра (рис. 9) [6—8].

Реализация специализированных ИС предполагает использование двух видов IP-ядер: мягкие (*Soft Cores*) и жесткие (*Hard Cores*). Первый вид ядер описывает структуру устройства или его части на уровне регистровых передач без физической привязки к топологии кристалла. IP-ядро интегрируют в общий проект и средствами САПР обеспечивают размещение и трассировку всего схем-

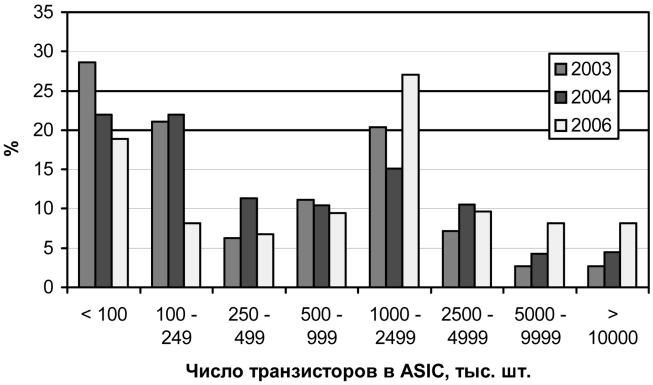


Рис. 6. Процентное соотношение сложности ASIC-проектов по годам

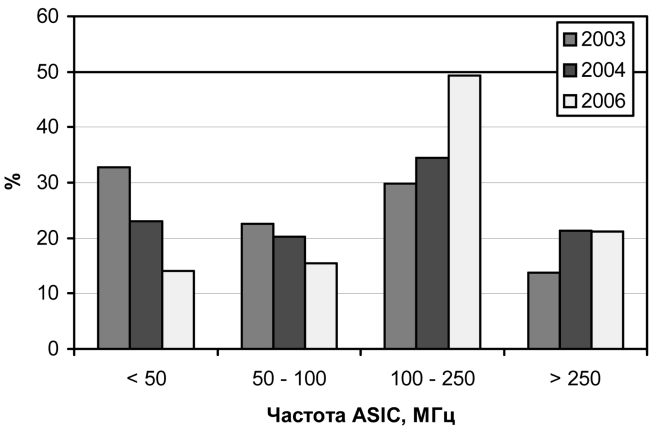


Рис. 7. Процентное соотношение объемов реализации ASIC-проектов по рабочей частоте

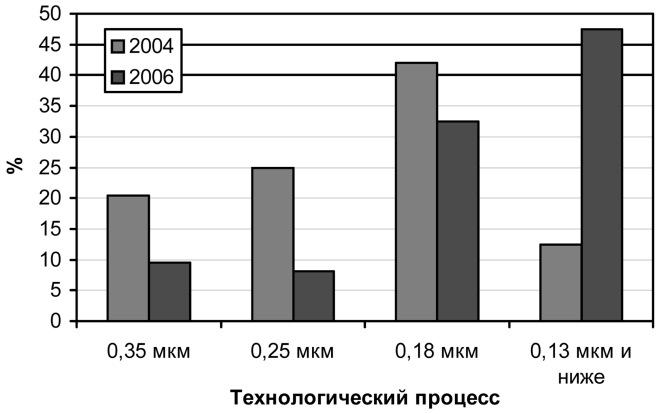


Рис. 8. Процентное соотношение используемых технологических процессов при проектировании ASIC

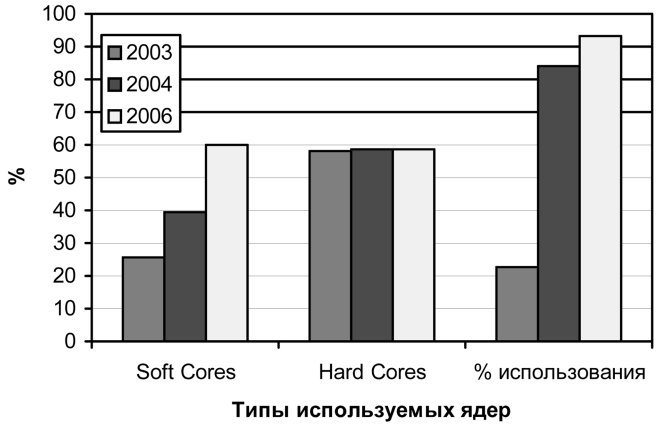


Рис. 9. Процентное соотношение используемых типов ядер при проектировании ASIC

ного решения на уровне топологии кристалла ИС. Второй вид ядер описывает многократно используемые библиотечные компоненты, реализованные на топологическом уровне с фиксированной привязкой к маскам фотошаблона. Жесткие ядра проходят полную верификацию и после их физической реализации в интегральном исполнении обеспечивают полное соответствие параметрам и характеристикам, указанным в спецификации.

Исторически жесткие ядра являются более распространенными при проектировании специализированных ИС по сравнению с мягкими ядрами. Однако в последнее время доля использования мягких ядер в ASIC-проектировании существенно возросла (рис. 9) [6]. Данную особенность можно объяснить развитием подсистем синтеза современных САПР, которые обеспечивают механизм однозначного перевода структурного описания функциональных блоков, представленных на уровне регистровых передач, в топологию кристалла ИС.

Высокие затраты на разработку специализированных ИС во многом связаны с необходимостью, как правило, неоднократного прототипирования устройства и его повторного перепроектирования, прежде чем удается обеспечить заданные в спецификации характеристики и параметры. Причем каждая такая итерация сопровождается реализацией высокотехнологичных и дорогостоящих операций, связанных с формированием топологии кристалла, изготовлением масок фотошаблонов, "выращиванием" микросхемы на специализированной фабрике с соблюдением строгих требований к производственному процессу и внутренней среде и др. Данная особенность определяет основные недостатки ASIC-проектов по сравнению с проектированием в базисе программируемых логических ИС (ПЛИС) — высокую стоимость и существенно высокие сроки разработки.

Проектирование на ПЛИС практически не связано с производственными затратами, поскольку основано на использовании стандартных микросхем. При выявлении несоответствия функционирования устройства вносят изменение в проект, синтезируют новый файл прошивки и загружают обновленную конфигурацию в ПЛИС. Данные операции не требуют сложных технических средств и специальных условий, поэтому могут быть выполнены даже в "домашних условиях". Отличительными особенностями проектов на ПЛИС по сравнению с ASIC являются высокая потребляемая мощность, низкая производительность и достаточно высокая стоимость микросхем. Однако сроки проектирования в базисе ПЛИС даже функционально сложных устройств принято исчислять неделями.

При реализации специализированных ИС обычно выполняют до четырех итераций получения прототипа, удовлетворяющего спецификации технического задания (рис. 10). На практике общее время проектирования ASIC зависит от нескольких факторов — числа инженеров, участвующих в проекте, и их квалификации, сложности реализуемого устройства, эффективности используемых САПР, применения IP-ядер при проектировании и др.

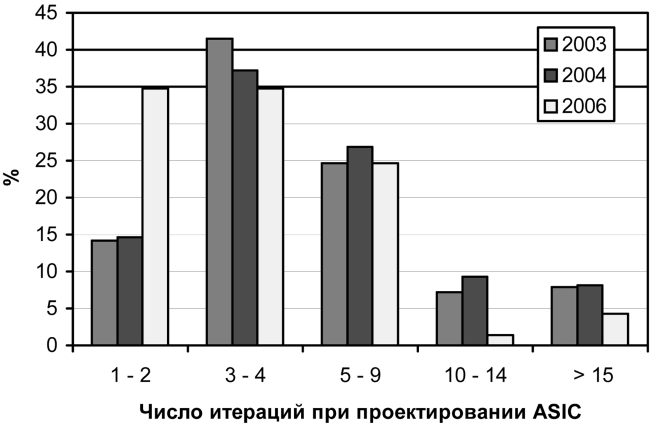
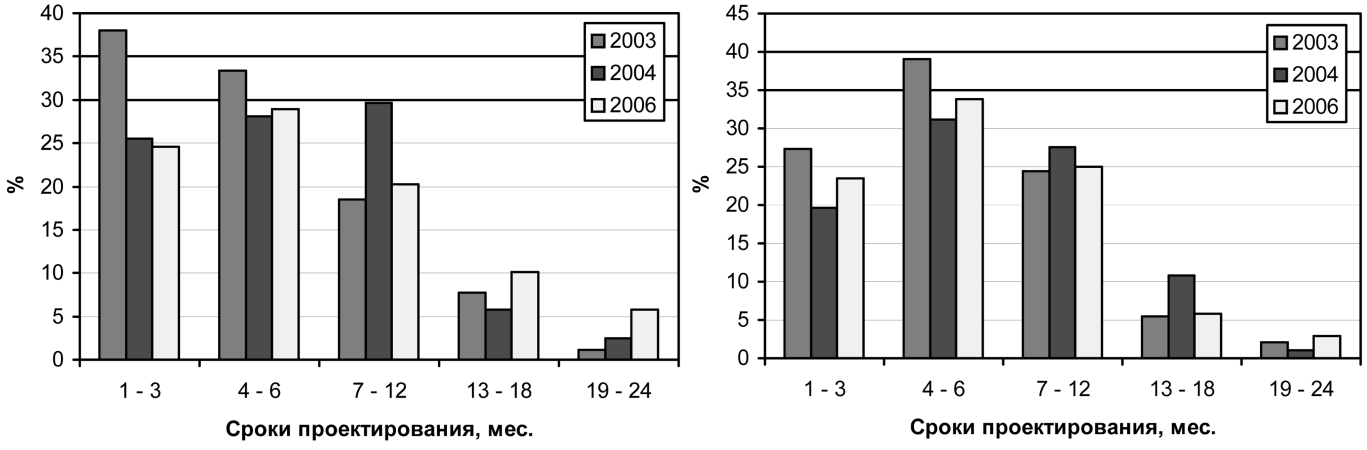


Рис. 10. Динамика циклов перепроектирования для ASIC



а) *б*)

Рис. 11. Динамика сроков проектирования ASIC:
а — этап "от технического задания до прототипа"; *б* — этап "от прототипа до готового изделия"

Полный цикл производства специализированной ИС принято разделять на два этапа. Первый из них носит название "***от технического задания до прототипа***" и включает все фазы проектирования от формирования спецификации на устройство до получения технологического файла в формате CIF (*Caltech Intermediate Form*) или GDSII (*Gerber Data Stream Information Interchange*). Данный файл необходим для изготовления микросхемы, он содержит описание топологии кристалла ИС в виде масок фотошаблонов. Общее время, необходимое для реализации этапа проектирования, как правило, занимает не более 12 месяцев. Однако в тех случаях, когда проектируют сложные устройства с повышенными требованиями к параметрам и характеристикам, данный процесс может занять и более года (рис. 11, *а*).

Второй этап называют "***от прототипа до готового изделия***". Он включает все фазы реализации конечной микросхемы, удовлетворяющей требованиям спецификации. Время реализации, как правило, соизмеримо со временем проектирования (рис. 11, *б*), но зависит не от сложности ASIC-проекта или квалификации инженеров-проектировщиков, а от особенностей технологического процесса производства ИС на конкретной кремниевой фабрике (*foundry*) [7, 8].

Современные тенденции развития рынка микроэлектроники показывают, что появление новых интегральных технологий стимулирует разработку современных методологий и маршрутов проектирования, а также средств САПР, которые обеспечивают возможность получать качественные решения за приемлемое время. Существующие технологии реализации ИС (ASIC или ПЛИС) обладают своими достоинствами и недостатками. На практике выбор технологии в первую очередь зависит от назначения проектируемого устройства. В настоящее время успех разрабатываемых изделий на рынке определяется не только стоимостными показателями, но и временны́ми. Одним из важнейших среди них является время выхода готового устройства на рынок (*time-to-market*). Качество готового изделия во многом зависит от используемых методов и подходов проверки его работоспособности.

Поиск и внедрение новых эффективных методов тестирования ИС, обеспечивающих получение результата за приемлемое время, является одним из активно развивающихся направлений современной микроэлектроники, которые имеют практическое значение.

Список литературы

1. **Global** Electronics: High Growth Products and New Markets // Electronics Industry Research and Knowledge Network, 2007.
2. **Asia-Pacific** IC Suppliers Marketshare to Reach 32 % in 2011 // Electronics Industry Research and Knowledge Network. 2007.
3. **Chipless** and Fabless Design Houses Still Prosper // Electronics Industry Research and Knowledge Network, 2007.
4. **European** Electronic Markets Forecast, 2007.
5. **International** Technology Roadmap for Semiconductors, 2007.
6. **Rusu S.** Trends and Challenges in VLSI Technology Scaling Towards 100 nm. Intel Corp. 2001. 46 p.
7. **Gartner** Dataquest and EE Times-Asia 2004 Report // Design Trends and EDA Tools. Mainland China and Taiwan, 2004.
8. **Gartner** Dataquest and EE Times-Asia 2006 Report // Design Trends and EDA Tools. Asia-Pacific, 2006.
9. **Chinnery D., Keutzer K.** Closing the Gap Between ASIC and Custom: an ASIC Perspective // Proc. of the 37th Design Automation Conference (DAC'00). 2000. P. 637—642.

УДК 621.396.6:658.512.011.56

Е. Н. Талицкий, д-р. техн. наук., проф.,
Владимирский государственный университет

Алгоритм проектирования виброзащиты электронной аппаратуры

Предлагается схема алгоритма проектирования виброзащиты электронной аппаратуры, впервые включающая практически все применяемые в настоящее время для этих целей способы. Предназначена для конструкторов электронной аппаратуры, применяемой на подвижных объектах.

Ключевые слова: *алгоритм, виброзащита, электронная аппаратура, виброизоляция, демпфирование, частотная отстройка.*

Введение

Методы защиты от вибраций электронной аппаратуры (ЭА), устанавливаемой на подвижных объектах, подразделяются на пассивные, обеспечивающие виброзащиту РЭС без дополнительных источников энергии, и активные, работающие только при дополнительном внешнем источнике энергии [1].

Активные виброзащитные устройства имеют значительно большие стоимость, массу, размеры и сравнительно низкую надежность. Поэтому для защиты ЭА от вибраций наиболее часто применяют пассивные методы виброзащиты. Они включают виброизоляцию, частотную отстройку, демпфирование колебаний, динамическое гашение колебаний и уменьшение виброактивности источника. Каждый из перечисленных способов имеет свои преимущества и недостатки.

Методы виброзащиты

Виброизоляция — метод вибрационной защиты посредством устройств, помещаемых между источником возбуждения и защищаемым объектом [1]. Действие виброизоляции сводится к ослаблению связей между источником и объектом. При этом уменьшаются динамические воздействия, передаваемые объекту. Применение виброизоляции для защиты аппаратуры в широком диапазоне частот вынужденных колебаний, как правило, не приводят к положительным результатам, так как эффективность виброизоляции на собственных частотах систем виброизоляции резко снижается [4].

Частотная отстройка применяется для устранения резонансных колебаний. Это обычно достигается за счет повышения собственных частот колебаний конструкции [2], которые должны не менее чем в 1,3 раз превышать частоты вынужденных колебаний. Частотную отстройку наиболее часто обеспечивают за счет увеличения жесткости конструкции путем уменьшения площади, повышения жесткости крепления, увеличения толщины и числа точек крепления основания ячейки, установки ребер жесткости и каркасов.

Этот способ целесообразно применять, когда диапазон частот действующих вибраций не превышает 400 Гц, в крайнем случае — 500 Гц. Было установлено, что при превышении этого диапазона устранить резонансные колебания невозможно без существенного, как правило, недопустимого, увеличения массы и габаритных размеров ячеек. Таким образом, частотная отстройка не решает задачу устранения резонансных колебаний ячеек аппаратуры, работающей при воздействии вибрации в широком диапазоне частот (свыше 500 Гц), особенно, если предъявляются жесткие требования к массе и габаритным размерам конструкции.

Для уменьшения амплитуд резонансных колебаний (АРК) в широком диапазоне частот используют методы *увеличения демпфирования* [3]. Следует отметить, что в типовых конструкциях ячеек демпфирование осуществляется за счет трех факторов: потерь энергии в окружающей среде, потерь в сочленениях и потерь в материале ячейки. Демпфирование за счет потерь в сочленениях и в материале ячейки называется конструкционным. Это демпфирование является доминирующим в ячейках ЭА. Практически только оно может ограничивать амплитуды резонансных колебаний до безопасных значений. Поэтому в конструкцию вводят дополнительные элементы с большим "внутренним трением", которые в несколько раз увеличивают потери энергии колебаний ячеек. При этом происходит снижение амплитуд колебаний в области резонанса. Такие элементы получили название полимерных демпферов (ПД). Применяемые в настоящее время для уменьшения АРК полимерные демпферы можно разделить на четыре вида: внешние и внутренние демпфирующие слои, демпфирующие ребра, демпфирующие вставки, динамические гасители колебаний с демпфированием.

Динамическое гашение вибрации заключается в присоединении к защищаемому объекту системы, реакции которой уменьшают размах вибрации объекта в точках присоединения этой системы [1]. Пассивные системы динамического гашения вибрации снижают амплитуды колебаний защищаемого объекта только на частоте собственных колебаний присоединенной системы. Поскольку для ЭА обычно характерна вибрация, как правило, в непрерывном и широком диапазоне частот,

применять динамические гасители колебаний (ДГК) для защиты такой аппаратуры нецелесообразно. Однако если ДГК обладает большими демпфирующими свойствами, он может в несколько раз уменьшить амплитуды резонансных колебаний и рассматриваться в этом случае как полимерный демпфер.

Иногда применяют *метод рационального размещения элементов*, заключающийся в расположении наиболее чувствительных к вибрации элементов в точках конструкции с наименьшими амплитудами колебаний, а также рациональной ориентации элементов относительно вектора вибрационных воздействий. Если элементы, размещаемые на ячейке, примерно равночувствительны к вибрации, то применять рассматриваемый метод не имеет смысла.

Другим недостатком является значительное увеличение сложности печатного монтажа, а следовательно, увеличение паразитных связей, приводящее к ухудшение радиочастотных свойств и увеличению стоимости аппаратуры. Сложность монтажа увеличивается за счет того, что элементы, имеющие наибольшее число связей, часто оказываются в противоположных концах ячейки. По этим причинам данный метод в рассматриваемый алгоритм не включен.

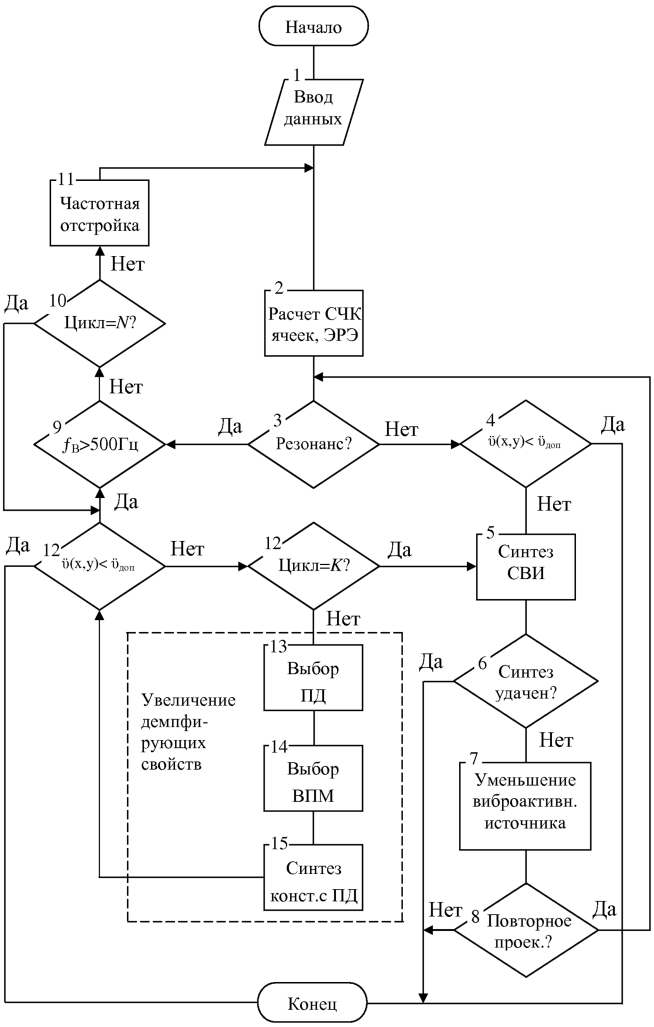
Очевидно, что многообразие способов виброзащиты требует разработки алгоритма их выбора.

Работа алгоритма

Результаты анализа отказов ЭА в условиях воздействия вибраций в широком диапазоне частот (до 500 Гц и выше) показывают, что наиболее часто отказывают электро- и радиоэлементы (ЭРЭ) вследствие недопустимо больших виброускорений и виброперемещений, возникающих при резонансных колебаниях ячеек, так как ячейки обладают, во-первых, небольшой изгибной жесткостью ячеек, выполняемых обычно на подложках из стеклотекстолита малой толщины (1—3 мм), а следовательно, имеющих низкие значения собственных частот колебаний (СЧК) (< 200—300 Гц), и во-вторых, широким диапазоном частот воздействующей вибрации (до 500 Гц и выше).

Поэтому первой, а часто и основной, задачей при обеспечении виброзащиты ЭА является устранение или уменьшение до допустимого уровня АРК ячеек.

Рассмотрим один из возможных вариантов схем проектирования, показанный на рисунке. Исходными данными (блок 1) являются диапазон частот и амплитуда действующей на ЭА вибрации, допустимые ускорения на ЭРЭ, конструктивные параметры блока и узлов — размеры, модули упругости материалов и т. д., условия эксплуатации, температурный диапазон и другие условия.



Алгоритм проектирования виброзащиты

В блоке 2 проводится расчет СЧК ячеек и ЭРЭ. Расчет может проводиться на основе аналитических или численных методов в зависимости от сложности конструкций. Если ячейки имеют прямоугольную форму, закреплены только по краям одним из "классических" способов и однородны, то предпочтение следует отдать аналитическим методам моделирования.

В блоке 3 проверяется возможность возникновения резонансных колебаний ячеек и ЭРЭ путем сравнения диапазона частот вибраций с собственными частотами колебаний элементов конструкций. Если резонансные колебания отсутствуют, то необходимо определить, не превышают ли виброускорения $\ddot{v}(x, y)$, действующие на ЭРЭ и заданные в ТЗ, допустимые виброускорения $\ddot{v}_{\text{доп}}$ для ЭРЭ (блок 4). Если не превышают, то никаких мер по виброзащите применять не нужно. В противном случае необходима виброизоляция в целях уменьшения виброускорений, действующих на

ЭА. В блоке 5 проводится синтез системы виброизоляции (СВИ). Он включает определение центра тяжести блока, расчет моментов инерции, выбор схемы расположения виброизоляторов и их статический расчет, выбор типа виброизоляторов, определения собственных частот блока на виброизоляторах и оценку эффективности системы виброизоляции [4]. Могут проводиться также расчеты на ударное воздействие и воздействие линейного ускорения. Если синтез удачен, т. е. удалось уменьшить виброускорения до допустимого уровня, то дальнейшие шаги по виброзащите не предпринимаются (блок 6). Если синтез неудачен, что может произойти вследствие жестких ограничений на размеры СВИ и очень низкой наименьшей частоты действующих вибраций (меньше 5 Гц), то необходимо уменьшить вибрации системы, в которых применяются ЭА. Если заказчик согласен на соответствующую корректировку ТЗ, то может проводиться повторное проектирование виброзащиты с учетом проведенной корректировки или использование уже найденного решения СВИ.

Если проверка на наличие резонансных колебаний показывает, что ячейки или другие элементы будут резонировать (блок 3), то необходимо применить частотную отстройку или увеличить демпфирующие свойства системы. Выбор одного из таких способов определяется двумя факторами. Во-первых, тем, что частотная отстройка, как правило, конструктивно-технологически выполняется проще, но при частотах возбуждения больше 500 Гц она приводит к значительному увеличению массы и габаритных размеров конструкции. Поэтому в блоке 9 проверяется, превышает ли верхняя частота возбуждения $f_в$ частоту, равную 500 Гц. Если нет, то для устранения резонансных колебаний целесообразнее применить частотную отстройку (блок 11). Для этого можно увеличить толщину конструкции, уменьшить площадь подложки, изменить способ крепления, применить ребро жесткости. В программе может быть реализовано несколько различных вариантов частотной отстройки. Выбор того или иного варианта может проводиться по различным критериям — стоимости, массе, габаритным размерам и др. После применения каждого варианта ведется расчет и проверяется условие отсутствия резонансных колебаний. Если за определенное число циклов N частотную отстройку провести не удается (блок 10), то необходимо попытаться решить задачу виброзащиты путем применения полимерных демпферов (ПД).

Поэтому в блоке 12 приводится сравнение допустимых значений виброускорений $\ddot{v}_{доп}(x, y)$ или виброперемещений $\ddot{v}(x, y)$ с полученными в блоке 3 значениями при резонансе ячейки. Допускаемые виброускорения обычно определяются из ТУ на ЭРЭ, допускаемые виброперемещения могут быть определены из дополнительного расчета по определению усталостной долговечности электрических выводов ЭРЭ.

Если виброускорения или виброперемещения не превышают допустимые значения, необходимо уменьшить АРК путем применения ПД (блок 13). Выбор типа ПД в настоящее время не формализован, поэтому обычно его проводят исходя из конструктивных особенностей ячейки этапа, на котором решается проблема виброзащиты и т. д. Выбор вибропоглощающего материала (ВПМ) (блок 14) проводится в зависимости от типа ПД, температурного и, в меньшей степени, частотного диапазонов. Так, например, ПД в виде внешнего слоя требует применения ВПМ с большим модулем упругости ($E > 10^8$ Па), а демпфирующие вставки выполняются из материалов с $E < 10^6$ Па. С учетом малой номенклатуры разработанных ВПМ может ставиться задача синтеза новых материалов. После выбора ВПМ оптимизируются параметры конструкции, обеспечивающие максимальное демпфирование и рассчитывается вибрационное поле ячейки (блок 15).

Если в течение заданного числа циклов K (блок 12) условие в блоке 12 выполняется, то задача виброзащиты решена. В противном случае необходимо дополнительно применить виброизоляцию (переход на блок 5). Если и в этом случае виброзащиту обеспечить не удается, необходимо принять меры по уменьшению виброактивности системы, в состав которой входит ЭА.

Заключение

На основе разработанного алгоритма создана программа [5], используемая в учебном процессе и в промышленности, позволяющая существенно сократить время проектирования виброзащиты электронной аппаратуры.

Список литературы

1. **Вибрации** в технике: Справочник: В 6 т. М.: Машиностроение, 1981. Т. 6. Защита от вибрации и ударов / Под ред. К. В. Фролова, 1981. 456 с.

2. **Токарев М. Ф., Талицкий Е. Н., Фролов В. А.** Механические воздействия и защита радиоэлектронной аппаратуры: Учеб. пособие для вузов / Под ред. В. А. Фролова. М.: Радио и связь, 1984. 224 с.

3. **Нашиф А., Джоунс Д., Хендерсон Дж.** Демпфирование колебаний: Пер. с англ. М.: Мир, 1988. 448 с.

4. **Ильинский В. С.** Защита РЭА и прецизионного оборудования от динамических воздействий. М.: Радио и связь, 1982. 296 с.

5. **Копылов И. А., Талицкий Е. Н., Шумарин С. В.** Свидетельство об официальной регистрации программы для ЭВМ № 2005611814 "Программный комплекс проектирования виброустойчивых электронных модулей", 2005.

УДК 004.056

И. В. Котенко, д-р техн. наук,
проф., вед. научн. сотр.,
В. В. Воронцов, аспирант,
А. А. Чечулин, аспирант,
А. В. Уланов, канд. техн. наук, мл. научн. сотр.,
Санкт-Петербургский институт информатики
и автоматизации РАН

Проактивные механизмы защиты от сетевых червей: подход, реализация и результаты экспериментов

Предлагается проактивный подход к защите от сетевых червей в сети Интернет, базирующийся на комбинировании различных механизмов обнаружения и сдерживания сетевых червей и автоматической динамической адаптации механизмов защиты в соответствии с текущей сетевой конфигурацией и сетевым трафиком. Описываются особенности данного подхода и программной реализации разработанной авторами системы моделирования механизмов защиты от сетевых червей. Приводятся результаты экспериментов, полученные при исследовании применения предлагаемого подхода для обнаружения и сдерживания как известных (CodeRed II, Slammer), так и потенциально возможных сетевых червей.

Ключевые слова: *сетевые черви, проактивный подход, механизмы обнаружения и сдерживания сетевых червей, моделирование, адаптация.*

Введение

В соответствии с опубликованной статистикой [1] рост числа компьютерных инцидентов за предыдущий год не уменьшился. Более половины из зарегистрированных происшествий было связано с вредоносным программным обеспечением. Учитывая динамику современных эпидемий сетевых червей и вирусов, можно отметить, что большую роль при ограничении размеров эпидемии играют механизмы обнаружения и сдерживания распространения сетевых червей и вирусов. Благодаря успешному обнаружению и сдерживанию сетевой эпидемии предоставляется время, необходимое для разработки и применения необходимых контрмер, например, анализа червя или вируса, выпуска соответствующего "патча", реконфигурации сети и др.

Предлагается проактивный подход к защите от сетевых червей, рассматриваются методика и среда исследования механизмов защиты на основе моделирования различных типов и экземпляров сетевых червей, а также механизмов защиты от них. Приводятся результаты моделирования представленных механизмов обнаружения и сдерживания.

Требования к реализации и сущность предлагаемого подхода

Предлагаемый подход предназначен для обнаружения сетевых червей (посредством выявления их действий по сканированию уязвимых хостов) и сдерживания их дальнейшего распространения за счет ограничения и блокирования посылаемых инфицированными узлами сетевых пакетов.

При разработке системы обнаружения и сдерживания были сформулированы следующие *основные требования* к разрабатываемым механизмам защиты от сетевых червей:

- *адекватность* — должны обеспечиваться низкие показатели пропуска атак и ложного срабатывания;
- *оперативность* — вредоносная сетевая активность должна обнаруживаться как можно раньше, данное требование напрямую влияет на ущерб, приносимый в результате эпидемии сетевых червей;
- *эффективность* использования системных ресурсов и возможность реализации на сетевом оборудовании;
- *автоматическое выполнение* — разрабатываемые механизмы должны функционировать без вмешательства (или при минимальном вмешательстве) администратора защиты;
- возможность обнаружения, кроме быстро сканирующих сетевых червей, также и червей, использующих скрытные алгоритмы сканирования.

Термин "проактивный" в контексте данной публикации означает механизм, "действующий до того, как ситуация станет критической". Свойству "проактивности" противопоставляется свойство "реактивности". "Реактивный" механизм определяется как "действующий после возникновения определенной ситуации". Предполагается, что *проактивное обнаружение и реагирование* против сетевых червей основывается на автоматических механизмах, которые используют информацию об "истории" анализируемых сетевых событий и про-

гнозе будущих событий. Такие меры используют автоматическую подстройку параметров обнаружения сетевых червей и ограничения трафика к текущему состоянию конфигурации сети и обрабатываемого трафика.

Данный подход базируется на выполнении двух основных функций:

- на комбинировании различных механизмов обнаружения и сдерживания сетевых червей (что позволяет объединить и усилить их достоинства и уменьшить недостатки);
- на автоматической динамической настройке (адаптации) основных параметров механизмов обнаружения и ограничения в соответствии с текущей сетевой конфигурацией и сетевым трафиком.

Для разработки проактивного подхода к обнаружению и сдерживанию сетевых червей предлагается использовать *комбинацию следующих особенностей* [2, 3]:

- "*многорезолюционный*" *способ* обнаружения и сдерживания сетевых червей, сочетающий использование нескольких интервалов времени ("окон") наблюдения сетевого трафика и применение различных порогов для отслеживаемых параметров;
- *использование различных алгоритмов и математических методов*, в том числе учитывающих события, характеризующие как аномальную, так и нормальную сетевую активность хоста;
- *многоуровневое комбинирование используемых алгоритмов* в виде системы базовых классификаторов, обрабатывающих данные о трафике, и метаклассификатора, осуществляющего выбор решения;
- *адаптация механизмов* обнаружения и сдерживания сетевых червей, заключающаяся в возможности изменять критерии обнаружения и сдерживания на основе статистических параметров сетевого трафика.

Особенности предлагаемого подхода

"Многорезолюционный" способ. Этот способ обнаружения и сдерживания сетевых червей позволяет сохранить свойства методов обнаружения на основе порогов, не требующих знания конкретных экземпляров атак. Он предоставляет возможности обнаружения, сопоставимые с сигнатурными методами, основанными на задании образцов конкретных экземпляров атак. Такой подход делает возможным обнаружение с низкой степенью ложных срабатываний не только агрессивных червей (червей, использующих для своего распространения сканирование с небольшой временной задержкой между посылкой сетевых пакетов), но и менее агрессивных червей.

В основе данного способа лежит следующее наблюдение [4]: во время функционирования неинфицированного хоста возможно изменение его сетевой активности, а именно — чередование периодов, когда хост передает большие объемы данных (или демонстрирует быстрый рост числа соединений с другими хостами), с периодами относительного бездействия. По этой причине, хотя на коротких интервалах времени "незараженные" хосты могут характеризоваться большим трафиком и большим числом соединений с новыми адресами назначения, эти хосты проявляют значительно более низкие средние значения интенсивности соединений при наблюдении в течение длительных интервалов времени.

Использование многорезолюционного способа с различными пороговыми значениями для разных интервалов времени может оказаться эффективным решением для обнаружения сетевых червей, использующих во время своего распространения различную скорость сканирования. Другой полезной для обнаружения возможностью является возможность обнаружения не только агрессивных сетевых червей, но и выявление некоторых методов скрытого сканирования. Таким образом, используя данный способ, можно обнаружить широкий спектр сетевых червей независимо от используемых сигнатур и стратегий сканирования, при одновременном сохранении легкости использования, свойственной алгоритмам, базирующимся на пороговых значениях.

Использование различных алгоритмов и математических методов и их многоуровневое комбинирование. Предлагаемый подход основывается на использовании комбинации механизмов обнаружения и сдерживания, основанных на разных алгоритмах и математических методах. Предполагается, что эти алгоритмы и методы могут учитывать события, характеризующие как аномальную, так и нормальную сетевую активность хостов. С одной стороны, такие алгоритмы могут использовать механизмы, базирующиеся на "вознаграждении". Для них при успешном установлении соединения увеличивается значение какого-либо счетчика (число кредитов, степени доверия к хосту и подобных им). С другой стороны, используются механизмы, основанные на учете тех или иных временных параметров. Например, признаком неудачного соединения может служить истечение таймаута соединения или ожидание ответа от узла более установленного времени.

В частности, предлагается использование следующих механизмов обнаружения и сдерживания, а также их модификаций:

- различные механизмы Virus throttling (например, для реализации на свитче (VT-S) [5] и на основе метода CUSUM (VT-C) [6];

- механизмы, основанные на анализе неудачных соединений (*Failed Connection Basic, FC-B*) [7];
- механизмы, базирующиеся на методе "порогового случайного прохождения" (*Threshold Random Walk, TRW*) [8];
- механизмы, основанные на методе упрощенного "порогового случайного прохождения" (*Simplified Threshold Random Walk, TRWS*) [9];
- механизмы, применяющие кредиты доверия (*Credit Based, CB*) [10];
- механизмы, основанные на DNS-статистике (*DNS*) [11].

Для обнаружения и сдерживания процесса распространения сетевых червей предполагается использовать двухуровневую архитектуру компонентов системы защиты. Ее первый уровень — основные классификаторы, которые реализуют отдельные методы (алгоритмы) обнаружения как одного класса (но с разными управляющими параметрами), так и разных классов. Второй уровень — метаклассификатор, осуществляющий комбинирование различных алгоритмов обнаружения и сдерживания, а также общие процедуры, необходимые для реализации при сетевой защите (например, поддержка списков контроля доступа для механизмов обнаружения и др.).

Адаптация механизмов обнаружения. Предполагается, что в процессе функционирования отслеживаются статистические параметры сетевого трафика. В зависимости от их значения или осуществляется выбор определенного механизма и его параметров, которые обеспечивают наилучшие показатели эффективности обнаружения и сдерживания в текущих условиях, или используется несколько механизмов, обладающих различными весами принимаемого решения о наличии сетевого червя. Таким образом, комбинирование данных механизмов заключается в выборе отдельных механизмов, которые наилучшим образом работают в текущих условиях, или в использовании нескольких механизмов (как различных классов, так и одного класса, но с различающимися параметрами), а также в формировании заключения о наличии сетевого червя на основе обработки заключений каждого механизма с различными весами.

В качестве отслеживаемых параметров используются, например, следующие величины:

- частота соединений (учитываются попытки соединения с новыми узлами);
- частота и число возникновения ошибочных соединений;
- частота и число "первоначальных" соединений;
- частота и число соединений с узлами, которые не содержатся в стеке (кэше) локального DNS-сервера и др.

Кроме того, для обнаружения и сдерживания сетевых червей должен поддерживаться ряд дополнительных механизмов. В их числе:

- отслеживание неиспользуемых адресов (*dark addresses*);
- использование списков контроля доступа (ACL-листов) для игнорирования соединений по определенным портам и протоколам;
- обработка сообщений от ложных (обманных) информационных систем (*honeypots*), например, могут блокироваться хосты, обращающиеся к таким системам, и ряд других.

Методика и среда исследования механизмов защиты

Для моделирования и оценки предлагаемого проактивного подхода была разработана методика и программная среда исследования (моделирования и анализа) механизмов обнаружения и сдерживания сетевых червей (рис. 1).

Суть предлагаемой *методики моделирования и анализа* сводится к проведению комплекса экспериментов на основе использования программной среды для различных значений входных параметров, измерению показателей эффективности механизмов защиты и их анализу.

Методика использует следующие *элементы программной среды*:

- источники трафика, формирующие нормальный трафик и трафик сетевого червя;
- анализатор трафика, предназначенный для синхронизации и стандартизации различных источников;
- контрольные задачи, включающие спецификацию сценариев тестирования, которые определяют способы использования среды моделирования и интерпретации результатов;
- библиотеки предобработки трафика и механизмов защиты от сетевых червей;

Рис. 1. Обобщенная архитектура среды исследования механизмов защиты

- модели средств тестирования, содержащие все элементы, необходимые для исследования реалистичных сценариев в программной среде, в том числе набор метрик (показателей) оценки, описывающих свойства исследуемых механизмов защиты.

Исследуются следующие *параметры эффективности* методов обнаружения и реагирования:
- доля блокированного и (или) задержанного легитимного трафика (степень ложных срабатываний, *false positives*);
- доля пропущенного злонамеренного трафика (степень пропусков атак, *false negatives*);
- время реакции на атаку.

Используется комбинированный *способ моделирования трафика*, заключающийся в применении в качестве входных данных различных записей реального трафика с дополнением их необходимым для исследования трафиком. В данном случае необходимый для исследования трафик — это трафик сетевых червей, как ранее известных, так и неизвестных, которые могут появиться в будущем, и трафик "быстрых" приложений, таких как P2P или NetBIOS/NS. Исследования могут проводиться как на записях реального трафика, так и на сгенерированном трафике с подключением различных модулей моделей трафика.

Основные компоненты реализованного генератора трафика:
- генератор нормального трафика;
- блок синхронизации нормального трафика и трафика червя;
- анализатор трафика;
- генератор трафика червя;
- фильтр трафика;
- блок задания начальных параметров фильтра.

Генератор нормального трафика реализует чтение записей трафика из файлов. *Блок синхронизации* нормального трафика и трафика червя служит для передачи сетевых пакетов механизмам защиты в упорядоченном по времени виде при одновременном использовании нескольких источников трафика. *Анализатор трафика* осуществляет выделение из записей трафика основных параметров, используемых для генерации трафика червя, а также служащих для исследования соответствия параметров трафика и механизмов обнаружения и реагирования, наилучшим образом функционирующих при данном трафике. *Генератор трафика червя* позволяет генерировать трафик как уже известных червей, так и неизвестных, которые могут появиться в будущем. Поскольку в записи трафика записаны ответы на запросы, блокированные методами реагирования, вводится *фильтр трафика*, который служит для имитации динамики в записи трафика. *Блок задания начальных параметров фильтра* позволяет использовать списки контроля доступа (*Access Control list, ACL*) для игнорирования соединений по определенным адресам, портам и протоколам.

Деятельность ранее известных червей имитируется путем задания в генераторе трафика предопределенных параметров их функционирования, соответствующих известным червям. *Действия новых червей* моделируются на основе задания произвольных параметров функционирования. Такими параметрами могут являться:
- тип соединения (TCP или UDP);
- частота генерации пакетов (число пакетов, генерируемых в секунду);
- изменение скорости, с которой проводится сканирование (она может быть постоянной или изменяться, в том числе случайным образом);
- тип сканирования или методика выбора адреса узла-получателя и порта (случайное, последовательное, на основе перестановок; частичное, локальное, топологическое, по хит-листам, комбинированное);

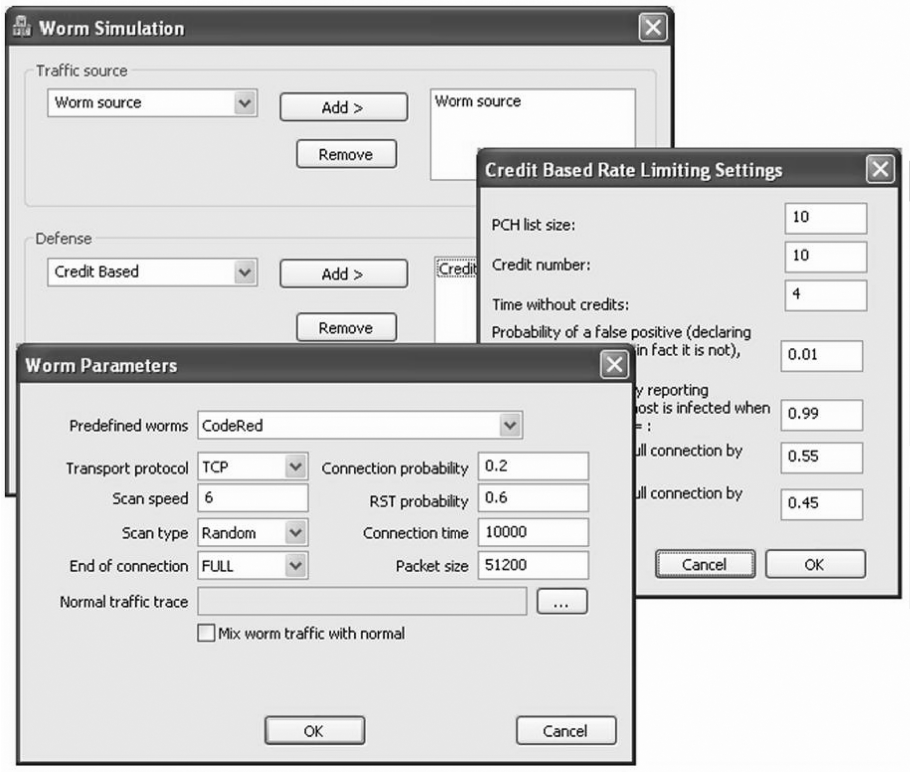


Рис. 2. Оконный интерфейс разработанной среды моделирования

- вероятность установления успешного TCP-соединения;
- размер пакета.

Пример *пользовательского интерфейса* разработанной программной среды моделирования представлен на рис. 2.

Основное окно пользовательского интерфейса разделено на три части. В верхней части, которая имеет название "Traffic source", пользователю предлагается специфицировать источники трафика для проведения моделирования. Средняя часть, которая имеет название "Defense", служит для специфицирования механизмов обнаружения и сдерживания сетевых червей для проведения моделирования. В нижней части находятся кнопки запуска сценария моделирования и выхода из программы.

Для проведения оценки механизмов обнаружения и сдерживания сетевых червей задаются различные *сценарии моделирования*. Сценарии включают набор экземпляров источников трафика и механизмов защиты. В результате моделирования определяются параметры эффективности механизмов защиты. После окончания моделирования генерируются отчеты о результатах эксперимента.

Разработано также специальное *консольное приложение*, которое предназначено для пакетного запуска большого числа экспериментов.

Результаты экспериментов

В данном разделе описываются результаты одного из комплексов проведенных экспериментов. Сравнительная оценка механизмов защиты от сетевых червей проводилась по результатам экспериментов на различных видах трафиков. Использовались трафики с низким и высоким процентом P2P-приложений. В дальнейшем трафики с низким процентом P2P-приложений называются *нормальными*, а трафики с высоким процентом P2P-приложений — *P2P-трафиками*.

Всего при моделировании было использовано семь нормальных трафиков, к трем из которых подмешивалось приложение-сканер, и пять P2P-трафиков. Все трафики смешивались с червями трех видов: CodeRed II, Slammer и искусственный червь (имитирующий один из возможных новых червей). Кроме того, механизмы защиты исследовались на всех указанных трафиках без примеси червя и на чистых трафиках червя.

В первую очередь используемые механизмы защиты проверялись на "чистом" трафике, а именно — на трафике, не содержащем трафик сетевых червей. Необходимость проведения экспериментов при отсутствии в сетевом трафике трафика червей обусловлена требованием минимизировать блокирование нормального сетевого трафика. Некоторые из полученных результатов приведены в табл. 1. В табл. 2 и табл. 3 представлены результаты экспериментов по обнаружению трафика червя CodeRed II в сети с нормальным и P2P-трафиком соответственно.

Таблица 1

Средние значения ошибок на трафиках без примеси трафика червей

№ п/п	Механизм	Среднее значение коэффициента ложных срабатываний на нормальном трафике	Среднее значение коэффициента ложных срабатываний на P2P-трафике
1	TRW	0,004300	0,002311
2	FC-B	0,005950	0,002946
3	CB	0,021800	0,080051
4	VT-S	0,022600	0,089913
5	VT-C	0,023400	0,000000

Таблица 2

Средние значения ошибок на нормальных трафиках, смешанных с трафиком червя CodeRed II

№ п/п	Механизм	Среднее значение суммы ошибок	Среднее значение коэффициента ложных срабатываний	Среднее значение коэффициента пропусков атак
1	TRW	0,069142	0,00498	0,064162
2	VT-S	0,032513	0,023277	0,009236
3	VT-C	0,033346	0,024118	0,009228
4	CB	0,027848	0,025692	0,002156
5	FC-B	0,071842	0,037267	0,034575

Таблица 3

Средние значения ошибок на P2P-трафиках, смешанных с трафиком червя CodeRed II

№ п/п	Механизм	Среднее значение суммы ошибок	Среднее значение коэффициента ложных срабатываний	Среднее значение коэффициента пропусков атак
1	CB	0,265824	0,265395	0,000429
2	VT-S	0,404845	0,208435	0,196409
3	VT-C	0,517375	0,067945	0,449429
4	FC-B	0,761482	0,031890	0,729592
5	TRW	0,790602	0,170161	0,620441

Заключение

В работе предложен проактивный подход к защите от сетевых червей, обладающий комбинацией следующих особенностей:

- "многорезолюционный" способ обнаружения и сдерживания сетевых червей, сочетающий использование нескольких интервалов времени наблюдения сетевого трафика и применение различных порогов для отслеживаемых параметров;

- использование различных алгоритмов и математических методов и их многоуровневое комбинирование в виде системы базовых классификаторов, обрабатывающих данные о трафике, и метаклассификатора, осуществляющего выбор решения;
- адаптивные механизмы, способные изменять критерии обнаружения на основе статистических параметров сетевого трафика.

Разработаны методика проведения исследований и архитектура программной среды исследования механизмов защиты. Выполнена реализация механизмов обнаружения и реагирования и средств их оценки. Разработана программная среда исследования механизмов защиты на основе моделирования.

Проведена большая серия экспериментов, использующих множество различных сценариев для различных значений входных параметров, подтверждающая эффективность предлагаемого подхода.

Эксперименты, проведенные над трафиками различных типов, позволяют сделать вывод о том, что универсального метода защиты от сетевых червей среди исследованных нет. В этих условиях возможны три направления улучшения защиты:
- выбор наилучшего механизма для данного вида трафика;
- комбинирование механизмов: принятие решения на основе голосования нескольких методов;
- совершенствование существующих механизмов.

Отметим, что во многих случаях для P2P-трафика предложенный в проекте механизм Virus throttling на основе метода CUSUM (VT-C) превзошел используемый в настоящее время на сетевых коммутаторах механизм VT-S. Этот факт свидетельствует о целесообразности его реализации на коммутаторах и использовании его вместо VT-S или совместно с VT-S в сетях с P2P-трафиком.

Основными направлениями дальнейших работ по реализации механизмов обнаружения и реагирования против сетевых червей являются следующие:
- совершенствование предложенных и реализованных механизмов защиты и подхода в целом;
- применение методов машинного обучения;
- реализация развитых механизмов адаптации;
- использование различных схем кооперации механизмов защиты;
- исследование возможных будущих червей и механизмов защиты от них.

Планируется дальнейшее развитие разработанного программного средства в целях создания интегрированной среды моделирования механизмов обнаружения и реагирования против сетевых червей.

Работа выполнена при финансовой поддержке РФФИ (проект № 07-01-00547), программы фундаментальных исследований ОИТВС РАН (контракт № 3.2/03), Фонда содействия отечественной науке, проектов Евросоюза POSITIF (контракт IST-2002-002314) и RE-TRUST (контракт № 021186-2) и других проектов.

Список литературы

1. **CSI/FBI 2006** Computer Crime and Security Survey. 2007.

2. **Котенко И. В., Воронцов В. В.** Проактивный подход к обнаружению и сдерживанию сетевых червей // Тр. Междунар. научно-техн. конф. "Интеллектуальные системы (AIS'07)" и "Интеллектуальные САПР (CAD-2007)". М.: Физматлит, 2007.

3. **Воронцов В. В., Котенко И. В.** Модели обнаружения и сдерживания сетевых червей на основе проактивного подхода // V Санкт-Петербургская межрегиональная конференция "Информационная безопасность регионов России (ИБРР-2007). Матер. конф. СПб, 2007. С. 47—48.

4. **Sekar V., Xie Y., Reiter M. K., Zhang H.** A Multi-Resolution Approach for Worm Detection and Containment // IEEE/IFIP DSN'06. 2006. P. 189—198.

5. **Sanchez M.** Virus Throttle as basis for ProActive Defense // Communications in Computer and Information Science (CCIS). Springer. 2007. Vol. 1. P. 57—74.

6. **Peng T., Leckie C., Kotagiri R.** Proactively Detecting DDoS Attack Using Source IP Address Monitoring // Networking 2004, Athens, Greece, May, 2004. Lecture Notes in Computer Science. 2004. V. 3042. P. 771—782.

7. **Chen S., Tang Y.** Slowing Down Internet Worms // 24th International Conference on Distributed Computing Systems (ICDCS '04), March 2004. IEEE Computer Society. 2004. P. 312—319.

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

9. **Weaver N., Staniford S., Paxson V.** Very fast containment of scanning worms, Revisited // Advances in Information Security. Malware Detection. Springer. 2007. V. 27. P. 113—145.

10. **Schechter S., Jung J., Berger A. W.** Fast Detection of Scanning Worm Infections // Proc. of the Seventh International Symposium on Recent Advances in Intrusion Detection, French Riviera, France, September 2004. Lecture Notes in Computer Science, Springer. 2004. V. 3224. P. 59—81.

11. **Wong C., Bielski S., Studer A., Wang C.** Empirical Analysis of Rate Limiting Mechanisms // 8th International Symposium on Recent Advances in Intrusion Detection (RAID 2005), Seattle, Washington, September 7—9, 2005. Lecture Notes in Computer Science, Springer. 2006. V. 3858. P. 22—42.

УДК 004.3

Н. А. Молдовян, д-р техн. наук, гл. науч. сотр.,
П. А. Молдовяну, канд. техн. наук,
начальник НТЦ,
Научный филиал ФГУП НИИ "Вектор"

Конечные группы векторов, содержащие подгруппы простого порядка большого размера

Рассматривается способ построения конечных нециклических групп векторов, содержащих подгруппы большого простого порядка. Нециклические подгруппы применены для синтеза алгоритмов электронной цифровой подписи.

Ключевые слова: *цифровая подпись, конечные группы, поля векторов.*

Введение

Придание юридической силы электронным документам основано на алгоритмах электронной цифровой подписи (ЭЦП) с открытым ключом. Такие алгоритмы основаны на вычислительно сложных задачах в конечных алгебраических структурах с ассоциативной операцией [1, 2]. В алгоритмах ЭЦП наиболее широко используется задача дискретного логарифмирования (ЗДЛ) в мультипликативных группах большого простого порядка, содержащихся в конечных простых $GF(p)$ или расширенных $GF(p^m)$, где $m \geqslant 2$, полях. При этом в качестве простого поля $GF(p)$ используется конечное кольцо $\mathbf{Z}_p$, где p — простое число большого размера, а в качестве расширенного поля $GF(p^m)$ — конечные поля многочленов. Для полей обоих типов были предложены методы решения ЗДЛ, имеющие субэкспоненциальную сложность, поэтому для обеспечения достаточной стойкости алгоритмов ЭЦП требуется использовать поля $GF(p)$ и $GF(p^m)$, размер порядка которых равен 1024 бит и более, что ограничивает производительность процедур аутентификации информации. Более высокую производительность обеспечивают алгоритмы ЭЦП, основанные на конечных группах точек эллиптической кривой (ЭК), групповой операцией в которых является операция композиции точек [3—5]. Однако в случае алгоритмов, использующих операции с точками ЭК, возрастание производительности ограничено тем, что операция композиции точек в качестве составного элемента включает вычисление обратных значений в конечном поле, над которым задана ЭК.

Недавно показано [6], что конечные группы и поля, формируемые в векторных пространствах со специально определенной операцией умножения, перспективны для разработки производительных алгоритмов ЭЦП. В [6] установлено, что при соответствующем выборе параметров векторного поля его мультипликативная группа содержит подгруппу простого порядка, размер которого близок к размеру порядка поля. Это позволяет использовать векторные поля с размером порядка 192—320 бит, благодаря чему при программной реализации алгоритмов ЭЦП может быть обеспечено повышение производительности в 5—10 раз по сравнению с алгоритмами ЭЦП на основе ЭК. При аппаратной реализации может быть обеспечено возрастание производительности в 10—50 раз за счет возможности распараллеливания процедур, требуемых для выполнения операции умножения векторов. Конечные группы векторов, рассмотренные в [6], которые возникают за пределами условий формирования векторных полей, являются нециклическими и содержат только подгруппы простого порядка относительно малого размера, что вносит ограничения на использование нециклических групп векторов в качестве примитива алгоритмов ЭЦП.

В настоящей статье рассматривается построение нециклических групп векторов, содержащих подгруппы большого простого порядка. Интерес к построению подгрупп большого простого порядка в рамках нециклических групп векторов состоит в том, что в последних сложность ЗДЛ представляется более высокой, поскольку входящие в нее векторы не могут быть представлены в виде степеней некоторого фиксированного вектора.

1. Определение операции умножения в конечном векторном пространстве

Конечное векторное пространство представляет собой множество наборов вида $(a_1, a_2, ..., a_m)$, включающих m элементов некоторого конечного поля, над которым задается векторное пространство. В соответствии с обозначениями [6] векторы будем записывать также и в виде суммы векторов-компонентов

$$a\boldsymbol{e} + b\boldsymbol{i} + ... + c\boldsymbol{j},$$

где $a, b, ..., c$ — координаты, являющиеся элементами конечного простого поля $GF(p)$; $\boldsymbol{e}, \boldsymbol{i}, ..., \boldsymbol{j}$ — базисные векторы. Операция сложения векторов определяется аналогично операции сложения многочленов степени $m - 1$ с помощью следующего простого соотношения:

$$(a\boldsymbol{e} + b\boldsymbol{i} + ... + c\boldsymbol{j}) + (a\boldsymbol{e} + b\boldsymbol{i} + ... + c\boldsymbol{j}) = \\ = (a + x)\boldsymbol{e} + (b + y)\boldsymbol{i} + ... + (c + z)\boldsymbol{j}.$$

В данной статье рассматривается построение нециклических мультипликативных групп в векторном пространстве, заданном над конечным кольцом $\mathbf{Z}_n$, где $n = kp$; k и p — различные простые числа. При заданном кольце $\mathbf{Z}_n$ может быть построено большое число различных групп векторов, различающихся вариантами задания операции умножения векторов. Операция умножения векторов определяется в соответствии с [6] по общему правилу умножения каждой компоненты первого вектора-сомножителя с каждой компонентой второго вектора-сомножителя. Возникающие произведения пар базисных векторов заменяются на вектор-компоненту $\varepsilon\boldsymbol{v}$, где $\varepsilon \in \mathbf{Z}_n$ и $\boldsymbol{v} \in \{\boldsymbol{e}, \boldsymbol{i}, ..., \boldsymbol{j}\}$. Конкретные виды операции умножения различаются конкретным правилом, описывающим такую замену.

Формальное представление умножения имеет вид

$$(a\boldsymbol{e} + b\boldsymbol{i} + ... + c\boldsymbol{j})(x\boldsymbol{e} + y\boldsymbol{i} + ... + z\boldsymbol{j}) =$$
$$= ax\boldsymbol{e}\cdot\boldsymbol{e} + ay\boldsymbol{e}\cdot\boldsymbol{i} + ... + az\boldsymbol{e}\cdot\boldsymbol{j} + bx\boldsymbol{i}\cdot\boldsymbol{e} + by\boldsymbol{i}\cdot\boldsymbol{i} + ... +$$
$$+ bz\boldsymbol{i}\cdot\boldsymbol{j} + ... + cx\boldsymbol{j}\cdot\boldsymbol{e} + cy\boldsymbol{j}\cdot\boldsymbol{i} + ... + cz\boldsymbol{j}\cdot\boldsymbol{j},$$

где вместо произведений $\boldsymbol{e}\cdot\boldsymbol{e}$, $\boldsymbol{e}\cdot\boldsymbol{i}$, $\boldsymbol{e}\cdot\boldsymbol{j}$, $\boldsymbol{i}\cdot\boldsymbol{e}$, $\boldsymbol{i}\cdot\boldsymbol{i}$, ...$\boldsymbol{i}\cdot\boldsymbol{j}$, ...$\boldsymbol{j}\cdot\boldsymbol{e}$, $\boldsymbol{j}\cdot\boldsymbol{i}$, ...$\boldsymbol{j}\cdot\boldsymbol{j}$ подставляются значения вида $\varepsilon\boldsymbol{v}$, задаваемые некоторой таблицей умножения базисных векторов. Конкретный вариант этой таблицы определяет конкретный тип операции векторного умножения. После выполнения указанной подстановки получается некоторая сумма однокомпонентных векторов, которые следует суммировать по правилу сложения векторов. Для синтеза алгоритмов ЭЦП требуются конечные векторные структуры с ассоциативной и коммутативной операцией умножения. Это условие реализуется путем использования таблиц, задающих коммутативное и ассоциативное умножение базисных векторов. В следующем разделе приводятся такие таблицы для различных значений размерности векторов m.

2. Правила умножения базисных векторов

Выполнение операции векторного умножения включает выполнение операций сложения и умножения (в конечном кольце $\mathbf{Z}_n$) над координатами векторов-сомножителей и над коэффициентами растяжения, присутствующими в таблицах умножения базисных векторов. Координаты и коэффициенты растяжения являются элементами конечного кольца $\mathbf{Z}_n$, где $n = kp$; k и p — различные простые числа.

Для случая $m = 3$ правила умножения базисных векторов зададим с помощью табл. 1, которая отличается от соответствующей таблицы, предложенной в [6], только тем, что присутствующие в ней коэффициенты растяжения рассматриваются как элементы кольца $\mathbf{Z}_n$. Легко показать, что табл. 1

Таблица 1

Правила умножения базисных векторов для случая $m = 3$

Базисные векторы	Базисные векторы		
	$\boldsymbol{e}$	$\boldsymbol{i}$	$\boldsymbol{j}$
$\boldsymbol{e}$ $\boldsymbol{i}$ $\boldsymbol{j}$	$\boldsymbol{e}$ $\boldsymbol{i}$ $\boldsymbol{j}$	$\boldsymbol{i}$ $\varepsilon\boldsymbol{j}$ $\varepsilon\mu\boldsymbol{e}$	$\boldsymbol{j}$ $\varepsilon\mu\boldsymbol{e}$ $\mu\boldsymbol{i}$

определяет операцию умножения базисных векторов, обладающую свойствами ассоциативности и коммутативности при произвольных значениях растягивающих коэффициентов $\varepsilon, \mu \in \mathbf{Z}_n$.

В случае размерности $m = 5$ имеется значительно больше вариантов задания операции векторного умножения. Разработка таблицы правил умножения базисных векторов для этого случая может быть выполнена по способу, предложенному в [7], который заключается в следующем. На первом шаге разрабатывается таблица без растягивающих коэффициентов. Она представляет собой таблицу Кэлли, задающую коммутативную групповую операцию над множеством базисных векторов. Затем в таблицу вносится один растягивающий коэффициент таким образом, чтобы операция, задаваемая таблицей, сохраняла свойства ассоциативности и коммутативности. Коммутативность обеспечивается построением таблицы, являющейся симметричной относительно главной диагонали. Сложнее реализуется свойство ассоциативности, однако для $m = 5$ эта задача легко решается переборным методом. После этого в 10 клеток таблицы вносится коэффициент растяжения $\varepsilon \in \mathbf{Z}_n$, причем распределение этого коэффициента по таблице вносится таким образом, чтобы сохранить свойства ассоциативности и коммутативности операции умножения базисных векторов. Затем находится другое распределение для коэффициента растяжения $\mu \in \mathbf{Z}_n$, потом — для $\tau \in \mathbf{Z}_n$ и для $\lambda \in \mathbf{Z}_n$. Для каждого из шести возможных вариантов исходных таблиц умножения базисных векторов возможны только четыре различных варианта распределения одного коэффициента растяжения.

После того как распределения для требуемого числа коэффициентов растяжения найдены, все эти распределения накладываются. Если в каких-то клетках таблицы присутствует произведение нескольких коэффициентов, то они могут быть перемножены, т. е. будет выполнена операция умножения в кольце $\mathbf{Z}_n$. Если такую процедуру выполнить в каждой клетке таблицы, то появится достаточно большое число различных растягивающих коэффициентов, однако всего имеется четыре независимых значений растягивающих коэффициентов. Два возможных варианта таблиц умножения базисных векторов в случае размерности $m = 5$ приведены в табл. 2 и 3.

Таблица 2

Умножение базисных векторов в пятимерном векторном пространстве (первый вариант)

БВ	Базисные векторы (БВ)				
	e	***i***	***j***	***k***	***u***
e	***e***	***i***	***j***	***k***	***u***
i	***i***	$\varepsilon\lambda\boldsymbol{j}$	$\varepsilon\tau\boldsymbol{k}$	$\varepsilon\lambda\boldsymbol{u}$	$\varepsilon\mu\tau\lambda\boldsymbol{e}$
j	***j***	$\varepsilon\tau\boldsymbol{k}$	$\varepsilon\tau\boldsymbol{u}$	$\varepsilon\mu\tau\lambda\boldsymbol{e}$	$\mu\tau\boldsymbol{i}$
k	***k***	$\varepsilon\lambda\boldsymbol{u}$	$\varepsilon\mu\tau\lambda\boldsymbol{e}$	$\mu\lambda\boldsymbol{i}$	$\mu\lambda\boldsymbol{j}$
u	***u***	$\varepsilon\mu\tau\lambda\boldsymbol{e}$	$\mu\tau\boldsymbol{i}$	$\mu\lambda\boldsymbol{j}$	$\mu\tau\boldsymbol{k}$

Таблица 3

Умножение базисных векторов в пятимерном векторном пространстве (второй вариант)

БВ	Базисные векторы (БВ)				
	e	***i***	***j***	***k***	***u***
e	***e***	***i***	***j***	***k***	***u***
i	***i***	$\tau\lambda\boldsymbol{u}$	$\varepsilon\mu\tau\lambda\boldsymbol{e}$	$\tau\lambda\boldsymbol{j}$	$\varepsilon\lambda\boldsymbol{k}$
j	***j***	$\varepsilon\mu\tau\lambda\boldsymbol{e}$	$\varepsilon\mu\boldsymbol{k}$	$\mu\tau\boldsymbol{u}$	$\varepsilon\mu\boldsymbol{i}$
k	***k***	$\tau\lambda\boldsymbol{j}$	$\mu\tau\boldsymbol{u}$	$\mu\tau\boldsymbol{i}$	$\varepsilon\mu\tau\lambda\boldsymbol{e}$
u	***u***	$\varepsilon\lambda\boldsymbol{k}$	$\varepsilon\mu\boldsymbol{i}$	$\varepsilon\mu\tau\lambda\boldsymbol{e}$	$\varepsilon\lambda\boldsymbol{j}$

Таблица 4

Правила умножения базисных векторов для случая $m = 8$

БВ	Базисные векторы (БВ)							
	e	***i***	***j***	***k***	***u***	***v***	***w***	***t***
e	***e***	***i***	***j***	***k***	***u***	***v***	***w***	***t***
i	***i***	$\lambda\varepsilon\boldsymbol{j}$	$\lambda\varepsilon\boldsymbol{k}$	$\tau\lambda\varepsilon\boldsymbol{u}$	$\varepsilon\boldsymbol{v}$	$\lambda\varepsilon\boldsymbol{w}$	$\lambda\varepsilon\boldsymbol{t}$	$\tau\lambda\varepsilon\mu\boldsymbol{e}$
j	***j***	$\lambda\varepsilon\boldsymbol{k}$	$\tau\lambda\varepsilon\boldsymbol{u}$	$\tau\varepsilon\boldsymbol{v}$	$\varepsilon\boldsymbol{w}$	$\lambda\varepsilon\boldsymbol{t}$	$\tau\lambda\varepsilon\mu\boldsymbol{e}$	$\tau\mu\boldsymbol{i}$
k	***k***	$\tau\lambda\varepsilon\boldsymbol{u}$	$\tau\varepsilon\boldsymbol{v}$	$\tau\varepsilon\boldsymbol{w}$	$\varepsilon\boldsymbol{t}$	$\tau\lambda\varepsilon\mu\boldsymbol{e}$	$\tau\mu\boldsymbol{i}$	$\tau\mu\boldsymbol{j}$
u	***u***	$\varepsilon\boldsymbol{v}$	$\varepsilon\boldsymbol{w}$	$\varepsilon\boldsymbol{t}$	$\tau\lambda\varepsilon\mu\boldsymbol{e}$	$\mu\boldsymbol{i}$	$\mu\boldsymbol{j}$	$\mu\boldsymbol{k}$
v	***v***	$\lambda\varepsilon\boldsymbol{w}$	$\lambda\varepsilon\boldsymbol{t}$	$\tau\lambda\varepsilon\mu\boldsymbol{e}$	$\mu\boldsymbol{i}$	$\lambda\mu\boldsymbol{j}$	$\lambda\mu\boldsymbol{k}$	$\tau\lambda\mu\boldsymbol{u}$
w	***w***	$\lambda\varepsilon\boldsymbol{t}$	$\tau\lambda\varepsilon\mu\boldsymbol{e}$	$\tau\mu\boldsymbol{i}$	$\mu\boldsymbol{j}$	$\lambda\mu\boldsymbol{k}$	$\tau\lambda\mu\boldsymbol{u}$	$\tau\mu\boldsymbol{v}$
t	***t***	$\tau\lambda\varepsilon\mu\boldsymbol{e}$	$\tau\mu\boldsymbol{i}$	$\tau\mu\boldsymbol{j}$	$\mu\boldsymbol{k}$	$\tau\lambda\mu\boldsymbol{u}$	$\tau\mu\boldsymbol{v}$	$\tau\mu\boldsymbol{w}$

Для случаев размерностей $m > 5$ сложность составления таблиц умножения базисных векторов с коэффициентами растяжения быстро возрастает. Для всех значений размерности имеется простой способ составления исходной таблицы, которая не включает коэффициентов растяжения. Этот способ состоит в последовательной записи строк таблицы, в которой каждая следующая строка получается из предыдущей путем циклического сдвига на одну клетку. При этом также имеются два простых и общих варианта распределения коэффициента растяжения. Один из этих вариантов состоит в выделении квадрата, состоящего из всех клеток, не входящих в первый ряд или в первую строку, и внесении коэффициента растяжения ε в клетки этого квадрата, расположенные на диагонали, идущей из правого верхнего угла в левый нижний угол, а также во все клетки, расположенные выше этой диагонали.

Второй вариант распределения состоит во внесении коэффициента растяжения μ во все клетки указанной диагонали и все клетки, расположенные ниже этой диагонали. Примеры реализации таких общих вариантов построения таблиц умножения базисных векторов представлены табл. 2 для случая $m = 5$ и табл. 4 для случая $m = 8$, при условии $\tau = \lambda = 1$.

Нахождение других вариантов распределения растягивающих коэффициентов имеет конкретную специфику для каждого значения размерности векторов. Нами найдены специфические распределения для случаев $m \in \{6, 7, 8, 10, 12\}$. Например, при $m = 8$ частные варианты представлены распределением растягивающих коэффициентов τ и λ в табл. 4. Видимо для случая $m = 8$ существуют и другие варианты частных видов распределений, отличные от четырех вариантов, представленных табл. 4. Из полученных результатов можно вывести частное правило, которому подчиняется каждое распределение растягивающих коэффициентов в случае простых значений m, согласно которому растягивающий коэффициент вносится в каждую клетку таблицы, где присутствует базисный вектор $\boldsymbol{e}$, кроме клетки, соответствующей произведению $\boldsymbol{e} \cdot \boldsymbol{e}$. Общим правилом являются запрет на внесение растягивающих коэффициентов в первую строку и в первый столбец таблицы.

3. Синтез групп, содержащих подгруппы с большим размером простого порядка

Рассмотрим подмножество $\{\boldsymbol{W}\}$ m-мерных ненулевых векторов, координаты которых являются элементами кольца $\mathbf{Z}_n$, где $n = kp$; k и p — простые числа, причем все растягивающие коэффициенты, за исключением ε, равны единице, а ε — невычет степени m по модулю k и по модулю p. При этом будем рассматривать случай, когда размерность делит значения $k - 1$ и $p - 1$. Операцию умножения векторов зададим по общей схеме, описанной в разделе 1 с использованием табл. 1—3. Множество векторов $\boldsymbol{W}$, каждому из которых можно сопоставить обратный вектор $\boldsymbol{W}^{-1}$, образует группу с операцией умножения векторов. Для синтеза алгоритмов ЭЦП представляет интерес построение групп, содержащих подгруппы большого простого порядка. Для определения условий образования таких групп векторов определим связь порядка Ω рассматриваемой группы векторов с размерностью и простыми числами k и p. Значение Ω равно мощности множества $\{\boldsymbol{W}\}$, которое можно найти, вычитая из мощности $\#\{\boldsymbol{U}\} = n^m = (kp)^m$ множества всех возможных векторов размерности m мощность $\#\{\boldsymbol{N}\}$ множества векторов, для которых не существует обратных значений.

Рассмотрим некоторый вектор $\boldsymbol{V} = (a, b, ..., c)$, принадлежащий множеству $\{\boldsymbol{W}\}$. Поскольку по предположению для $\boldsymbol{V}$ существует обратный вектор, то векторное уравнение

$$\boldsymbol{V} \cdot \boldsymbol{X} + \boldsymbol{E}, \qquad (1)$$

где $\boldsymbol{E} = (1, 0, ..., 0)$, имеет решение относительно вектора $\boldsymbol{X} = (x, y, ..., z)$ как неизвестной величины. Указанному векторному уравнению соответствует система сравнений по модулю $n = kp$ следующего вида:

$$\begin{cases} f_{11}(a, b, ..., c)x + f_{12}(a, b, ..., c)y + ... + \\ + f_{13}(a, b, ..., c)z \equiv 1 \bmod n; \\ f_{21}(a, b, ..., c)x + f_{22}(a, b, ..., c)y + ... + \\ + f_{23}(a, b, ..., c)z \equiv 0 \bmod n; \\ ..... \\ f_{31}(a, b, ..., c)x + f_{32}(a, b, ..., c)y + ... + \\ + f_{33}(a, b, ..., c)z \equiv 0 \bmod n, \end{cases} \quad (2)$$

где коэффициенты f_{kl} при неизвестных $x, y, ..., z$ определяются коэффициентами растяжения и координатами вектора $\boldsymbol{V}$, а неизвестными являются координаты вектора $\boldsymbol{X} = (x, y, ..., z)$. В силу разрешимости векторного уравнения (1) значения координат $x, y, ..., z$ удовлетворяют системе сравнений (2).

Эта же система, записанная по модулю k, имеет вид

$$\begin{cases} f_{11}(a, b, ..., c)x + f_{12}(a, b, ..., c)y + ... + \\ + f_{13}(a, b, ..., c)z \equiv 1 \bmod k; \\ f_{21}(a, b, c)x + f_{22}(a, b, ..., c)y + ... + \\ + f_{23}(a, b, ..., c)z \equiv 0 \bmod k; \\ ..... \\ f_{31}(a, b, c)x + f_{32}(a, b, ..., c)y + ... + \\ + f_{33}(a, b, ..., c)z \equiv 0 \bmod k. \end{cases} \quad (3)$$

Эта же система, записанная по модулю p, имеет вид

$$\begin{cases} f_{11}(a, b, ..., c)x + f_{12}(a, b, ..., c)y + ... + \\ + f_{13}(a, b, ..., c)z \equiv 1 \bmod p; \\ f_{21}(a, b, c)x + f_{22}(a, b, ..., c)y + ... + \\ + f_{23}(a, b, ..., c)z \equiv 0 \bmod p; \\ ..... \\ f_{31}(a, b, c)x + f_{32}(a, b, ..., c)y + ... + \\ + f_{33}(a, b, ..., c)z \equiv 0 \bmod p. \end{cases} \quad (4)$$

Если система (2) имеет решение, то имеют решения и системы (3) и (4), т. е. разрешимость систем (3) и (4) является необходимым условием разрешимости системы (2). Для определения значения $\#\{\boldsymbol{N}\}$ рассмотрим векторы $\boldsymbol{V}'$ и $\boldsymbol{V}''$, получаемые из векторов $\boldsymbol{V}$ по формулам

$$\boldsymbol{V}' = (a \bmod k, b \bmod k, ..., c \bmod k);$$

$$\boldsymbol{V}'' = (a \bmod p, b \bmod p, ..., c \bmod p),$$

где $\boldsymbol{V} = (a, b, ..., c)$. Если в системах (3) и (4) координаты вектора $\boldsymbol{V}$ заменить на координаты векторов $\boldsymbol{V}'$ и $\boldsymbol{V}''$ соответственно, то каждое из сравнений останется равносильным исходному виду. Поскольку $m|k - 1$, $m|p - 1$ и коэффициент ε непредставим в виде квадрата в полях $GF(k)$ и $GF(p)$, то согласно результатам работы [6], векторы $\boldsymbol{V}'$ и $\boldsymbol{V}''$ принадлежат векторным полям, заданным над простыми полями $GF(k)$ и $GF(p)$ соответственно. После указанной замены каждая из систем (3) и (4) не имеет решения только для нулевого вектора $(0, 0, ..., 0)$ в соответствующем векторном поле. Таким образом, для наборов из m чисел вида $\boldsymbol{N}' = (h_1k, h_2k, ..., h_mk)$, где $h_1, h_2, ..., h_m \in \{0, 1, 2, ..., p - 1\}$, и $\boldsymbol{N}'' = (h_1'p, h_2'p, ..., h_m'p)$, где $h_1', h_2', ..., h_m' \in \{0, 1, 2, 3, ..., k - 1\}$, системы (3) и (4), соответственно, являются неразрешимыми. С учетом указанного выше необходимого условия разрешимости системы (2) последнее означает, что для векторов вида $\boldsymbol{N}'$ и $\boldsymbol{N}''$ система (2) не имеет решений, т. е. для таких векторов не существует обратных значений в векторном пространстве, заданном над кольцом $\mathbf{Z}_n$. Следовательно, имеем

$$\#\{\boldsymbol{N}\} = \#\{\boldsymbol{N}'\} + \#\{\boldsymbol{N}''\} = p^m + k^m - 1.$$

Значение порядка группы m-мерных векторов, заданной над кольцом $\mathbf{Z}_n$, равно

$$\Omega = \#\{\boldsymbol{W}\} - \#\{\boldsymbol{N}\} = (kp)^m - p^m - k^m + 1 = \\ = (k^m - 1)(p^m - 1).$$

Полученная формула позволяет определить значения простых порядков подгрупп, которые всегда являются циклическими. Вопрос о том, является ли построенная группа векторов порядка Ω циклической или не является, решается экспериментом. Опыт показал, что во всех случаях формируются нециклические группы, максимальный порядок которых не превосходит наименьшее общее кратное значений $k^m - 1$ и $p^m - 1$. Теоретически определенное значение $\Omega = (k^m - 1)(p^m - 1)$ подтверждается вычислительным экспериментом — все делители циклических подгрупп векторов являются также делителями и числа Ω.

4. Выбор параметров групп векторов для синтеза алгоритмов ЭЦП

Для разработки алгоритмов ЭЦП следует выбирать такие значения m, k и p, при которых значение Ω будет содержать большой простой делитель, размер которого близок к значению $|\Omega|$, где $|\Omega|$ обозначает битовую длину (размер) числа Ω. В связи с этим целесообразно выбирать значения k, имеющие минимальный размер, при котором можно обеспечить выполнение условия $m|k - 1$. Значение p следует выбирать с учетом задания требуемого размера простого порядка циклической подгруппы при выполнении условия $m|p - 1$. Наибольшие значения размера простого множителя числа $p^m - 1$ при заданном размере $|p|$ могут быть получены при простом значении размерности m. Действительно, только при прос-

Таблица 5

Значения простого числа p, обеспечивающие формирование подгрупп большого простого порядка для случая простых значений m

m	p	$q = m^{-1}(p^{m-1} + p^{m-1} + ... + p + 1)$
3	48957116200618261	7989330755602796800651182824274 61
5	835951676471	97668538258277323781238734252046720316673459741
7	566166749	47050919878490057384987147439163071988383749426338 93
11	177013	27457525370838803905620570394966352088511358055027 73
13	16693	36015992367075740098701090168244788260128296735577

Таблица 6

Примеры значений простого числа p, обеспечивающие формирование подгрупп большого простого порядка для случая четных значений m

m	p	$q' = 0{,}5(p^{m/2} + 1)$
2	30367255363837342851191952040227885 8821	15183627681918671425595976020113942 9411
4	12947807173190922719	83822855297067156515075089129315176481
8	7364803549	14710067706001738054286271651455735754 01

тых значениях размерности $m|p-1$ значение p можно выбрать таким образом, что множитель $q = m^{-1}(p^{m-1} + p^{m-1} + ... + p + 1)$, содержащийся в разложении числа p^{m-1}, будет простым. Если некоторое простое число q делит порядок конечной группы, то, согласно теореме Силова [8], в этой группе содержится подгруппа простого порядка q. Некоторые конкретные варианты таких значений p, задающих простое значение q, приведены в табл. 5. Таким образом, при соответствующем выборе числа p заданного размера в построенных группах векторов содержится подгруппа простого порядка размера $|q| = (m-1)|p| - |m| \approx \approx (m-1)|p|$. Для построения алгоритмов ЭЦП требуется использовать подгруппы простого порядка Q, размером не менее $|Q| \geqslant 160$ бит. Для этой цели следует формировать поля m-мерных векторов, заданных над простым полем с размером характеристики $|p|$, удовлетворяющим условию

$$|p| \geqslant \frac{|Q| - |m|}{m-1} \approx \frac{|Q|}{m-1} \text{ (бит).}$$

Интересными для синтеза алгоритмов ЭЦП представляются случаи размерности $m \approx \{3, 5, 7, 11, 13, 17, 19\}$. Для четных значений размерности интерес представляют значения m, представимые в виде натуральной степени числа 2, и случай $m = 8$, когда размер максимального простого делителя $q' = 0{,}5(p^{m/2} + 1)$ числа $p^m - 1$ равен $|q'| \approx \approx (m/2)|p|$ (см. табл. 6). В остальных случаях размер этого делителя значительно меньше, поскольку выражение $p^{m/2} + 1$ при других значениях m разлагается на нетривиальные множители.

Синтез алгоритмов ЭЦП на основе разработанных нециклических групп многомерных векторов состоит в выборе параметров этой группы, при которых обеспечивается формирование циклической подгруппы простого порядка с заданным значением размера. Затем задаются конкретные процедуры генерации ЭЦП и проверки ее подлинности. Указанные процедуры могут быть составлены по аналогии с большим числом известных схем ЭЦП.

Рассмотрим возможный вариант алгоритма ЭЦП, использующего вычисления в нециклической группе векторов.

Зададим группу Г пятимерных векторов с параметрами $k = 101$ и $p = 18060648844193430701$. Операцию умножения определим по табл. 2 при значениях растягивающих коэффициентов $\varepsilon = 1111111111113333333$ и $\mu = \tau = \lambda = 1$. При выбранном значении простого числа p группа Г содержит циклическую подгруппу простого порядка, имеющего значение $q = 2127959657873\backslash\backslash$ $72510032370340113348949764553959999474202 89\backslash\backslash$ $7441615478322548414 01$* и размер $|q| \geqslant 240$ бит. В качестве генератора G данной подгруппы можно взять вектор

$$\boldsymbol{G} = 242299117982852860829\boldsymbol{e} + 64815356422215586784 6\boldsymbol{i} + 88786584670953\backslash\backslash 144633\boldsymbol{j} + 1978805266047532139 71\boldsymbol{k} + 9894404238787573886 21\boldsymbol{u}.$$

Проверка показывает, что, действительно, имеем $\boldsymbol{G}^q = \boldsymbol{E}$.

Циклическая подгруппа, генерируемая вектором $\boldsymbol{G}$, может быть положена в основу следующей схемы ЭЦП. Открытый ключ $\boldsymbol{Y}$ формируется в виде вектора $\boldsymbol{Y} = \boldsymbol{G}^x$, а ЭЦП в виде пары чисел (h, s) вычисляется по подписываемому сообщению M следующим образом:

1) выбрать случайное число $t < q$ и вычислить вектор $\boldsymbol{R} = \boldsymbol{G}^t$;

2) используя некоторую специфицированную хэш-функцию F_h, вычислить хэш-код h от сообщения M с присоединенным к нему вектором $\boldsymbol{R}$: $h = F_h(M, \boldsymbol{R})$;

3) вычислить значение $s = t - xh \bmod q$.

* Знак \\ обозначает перенос записи числа на другу строку.

Проверка подлинности подписи (h, s) выполняется по следующему алгоритму:

1) вычислить вектор $\boldsymbol{R}' = \boldsymbol{Y}^h \cdot \boldsymbol{G}^s$;

2) вычислить значение $h' = F_h(M, \boldsymbol{R}')$;

3) сравнить значения h' и h; если $h' = h$, то ЭЦП является подлинной.

Заключение

Предложен способ построения нециклических групп векторов, содержащих подгруппы простого порядка большого размера. Особенность построенной группы векторов состоит в том, что их координаты являются элементами конечного кольца $\mathbf{Z}_{kp}$, где k, p — простые числа, причем k — число сравнительно малого размера, а растягивающие коэффициенты являются квадратичными невычетами по модулю k и по модулю p. Путем выбора соответствующих конкретных значений p обеспечивается размер подгруппы простого порядка, равный приблизительно значению $(m - 1)|p|$. Благодаря тому, что векторное умножение является свободным от операции инверсии в конечном поле, над которым определяются группы векторов, на основе построенных алгебраических структур могут быть разработаны алгоритмы ЭЦП, обладающие более высокой производительностью по сравнению с алгоритмами на основе ЭК. Актуальной задачей дальнейшего исследования в данном направлении является исследование зависимости сложности ЗДЛ от коэффициентов растяжения и размера простого числа k.

Работа поддержана грантом РФФИ № 08-07-00096-а.

Список литературы

1. **Pieprzyk J., Hardjono Th. and Seberry J.** Fundamentals of Computer Security. Berlin: Springer-Verlag. 2003. 677 p.

2. **Menezes A. J., Van Oorschot P. C. and Vanstone S. A.** Handbook of Applied Cryptography. CRC Press, Boca Raton, FL. 1997. 780 p.

3. **Болотов А. А., Гашков С. Б., Фролов А. Б., Часовских А. А.** Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы. М.: КомКнига, 2006. 324 с.

4. **Болотов А. А., Гашков С. Б., Фролов А. Б.** Элементарное введение в эллиптическую криптографию. Протоколы криптографии на эллиптических кривых. М.: КомКнига, 2006. 274 с.

5. **Menezes A. J. and Vanstone S. A.** Elliptic Curve Cryptosystems and Their Implementation // Journal of cryptology. 1993. Vol. 6. N 4. P. 209—224.

6. **Молдовян Н. А.** Группы векторов для алгоритмов электронной цифровой подписи // Вестник СПбГУ. Сер. 10. 2008.

7. **Молдовян Н. А.** Алгоритмов аутентификации информации в АСУ на основе полей векторов // Автоматика и телемеханика. 2008.

8. **Каргаполов М. И., Мерзляков Ю. И.** Основы теории групп. М.: Физматлит, 1996. 287 с.

УДК 004.056.53

М. В. Бочков, д-р техн. наук, проф.,
Академия ФСО РФ,
А. А. Шкадов, ст. инженер,
НИИ "Энергия" ФСО РФ

Формальная модель состояний системы защиты компьютерной сети при использовании политик информационной безопасности

Развивается подход к управлению защищенностью компьютерной сети на основе политик безопасности. Предложена модель определения состояния защищенности компьютерной сети путем формирования наборов параметров безопасности для каждого уровня защиты сети.

Ключевые слова: *политика безопасности, уровень защищенности, уязвимости, параметры конфигурации сети.*

Анализ прецедентов нарушения информационной безопасности показывает, что большинство из них происходит вследствие отсутствия контроля над состоянием политики безопасности информационной системы. В данных условиях перспективным направлением исследований в области управления защитой компьютерных сетей является объединение задач оценки защищенности и управления параметрами системы защиты информации (СЗИ). Это позволяет в условиях изменения характера воздействий нарушителя поддерживать защищенность компьютерной сети на максимально возможном уровне за счет оперативного реагирования на угрозы нарушения информационной безопасности путем динамической смены политик безопасности.

Подходы к управлению защитой компьютерной сети

Анализ [1, 2] существующих решений показывает, что на сегодняшний день можно выделить два основных подхода к управлению защитой ресурсов компьютерной сети в условиях локальных или сетевых атак:

- безопасность на основе распространяемых политик;
- предотвращение вторжений путем нейтрализации уязвимостей.

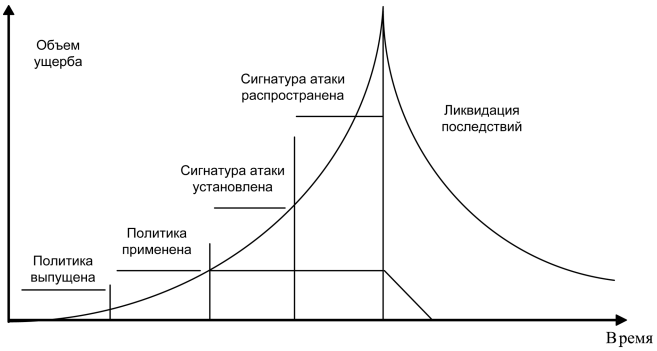


Рис. 1. Механизм использования политики безопасности

В рамках первого подхода пользователь получает набор политик (ограничений), которые позволяют защититься от новой угрозы до появления сигнатур, однозначно идентифицирующих соответствующую атаку, или обновлений, исключающих последнюю. Механизм использования политики безопасности представлен на рис. 1. Данный подход получил развитие в рамках технологии антивирусной защиты, разработанной корпорацией Trend Micro [6].

Отличительной особенностью второго подхода является предупреждение локальных и удаленных атак путем анализа журналов регистрации событий на территориально распределенных узлах компьютерной сети и реагирования на них в реальном времени. Управление защитой реализуется путем активизации набора правил, соответствующего признакам атаки. Правила представляют собой логические выражения, построенные на предикатах первого порядка. Предикаты используются для определения условий возникновения отслеживаемых ситуаций, а логические выражения определяют способы реагирования в зависимости от истинности или ложности предикатов. В рамках защиты сетевого периметра от внешних злоумышленников данный подход предусматривает использование технологии предотвращения вторжений, которая позволяет избежать попадания на компьютер вирусов и червей, а также защититься от хакерских атак. Достигается это посредством своевременной блокировки отдельных портов (например, тех, через которые на компьютер попадает опасный на данный момент червь), запрета доступа к определенным файлам и папкам.

Перечисленные выше подходы имеют ряд существенных недостатков:

- при первом подходе защищенность сети ставится в зависимость от качества услуг внешнего сервиса. Также возникает проблема смены политик вследствие их большого числа;
- недостатком второго подхода является отсутствие возможности обеспечения баланса между защищенностью и функциональностью компьютерной сети в условиях воздействий злоумышленника;
- отсутствие полного цикла оперативного управления параметрами СЗИ;
- большое число ложных срабатываний.

На основе вышеизложенного можно сделать вывод об актуальности создания модели, согласно которой в зависимости от состояния системы защиты путем изменения параметров защищенности сети устанавливалась бы необходимая политика информационной безопасности. Начальным этапом в построении такой модели является формализация состояния защищенности сети.

Формализация состояния защищенности сети при использовании политик безопасности

Для сокращения размерности задачи управления и упрощения формализации предлагается рассматривать систему защиты компьютерной сети в виде совокупности взаимодействующих подсистем. Декомпозицию системы защиты (рис. 2) проведем исходя из предпосылки, что невозможно обеспечить требуемый уровень защищенности ресурсов компьютерной сети только с помощью одного отдельного средства или с помощью их простой совокупности. Необходимо их системное согласование между собой. В этом случае реализация любой угрозы возможна только в случае преодоления всех установленных уровней защиты.

В данном случае под угрозой будем понимать событие, реализация которого способна нанести ущерб защищаемому объекту путем воздействий на его компоненты.

Уровни защиты	Средства защиты
Периметр	Межсетевые экраны, анализаторы для шлюзов, прокси, анализаторы контента, средства построения виртуальных частных сетей, системы предотвращения вторжений
Сеть	Сетевые системы обнаружения атак; системы предотвращения вторжений, межсетевые экраны, сетевые сканеры, средства аутентификации, средства управления доступом, средства управления политиками безопасности
Сервер	Узловые системы обнаружения атак, системные сканеры, анализаторы политик безопасности, антивирусы, средства управления доступом, средства аутентификации
Приложения	Средства контроля ввода данных, анализаторы политик безопасности, антивирусы, средства контроля доступа, средства аутентификации
Данные	Шифрование, подпись, управление доступом, аутентификация

Рис. 2. Декомпозиция системы защиты

Под политикой безопасности будем понимать множество установленных на некотором интервале функционирования компьютерной сети параметров безопасности соответствующих уровней системы защиты сети.

Разработанная в результате исследований математическая модель базируется на следующих множествах:

T — множество угроз;

V — множество уязвимостей;

M — множество политик безопасности;

P — множество параметров безопасности средств защиты соответствующих уровней;

U — множество уровней защиты.

Элементы модели определяются в терминах трех категорий: субъектов, объектов и воздействий первых на вторые.

На основании вышеизложенного предложено представление множества M в виде совокупности классов политик безопасности:

- Политики безопасности периметра;
- Политики безопасности сети;
- Политики безопасности серверов;
- Политики безопасности приложений;
- Политики безопасности данных.

Для описания взаимосвязи между элементами множеств M, P и U будем использовать математический аппарат, представленный в [5]. Определим тернарное отношение W на множестве $K = M \times P \times U$.

Принадлежность элемента (m, p, u) отношению W, где $m \in M$, $p \in P$ и $u \in U$, интерпретируется следующим образом.

1. Каждому уровню защиты u соответствует определенный набор параметров безопасности p, которые составляют политику информационной безопасности m.

2. С каждым уровнем защиты $u_i \in U$ где $i = 1, 2, ..., n$, связано множество p_i, являющееся подмножеством множества P, которое включает в себя параметры защищенности. При этом один уровень защиты u_i не может включать в себя все параметры защищенности множества P и, вместе с тем, не существует уровня защиты, с которым не был бы связан ни один из параметров безопасности.

3. С каждым параметром безопасности $p_i \in P$ связано множество u_i, являющееся подмножеством множества U и включающее в себя уровни защиты. При этом один параметр безопасности p_i не может входить во все уровни защиты, и, наоборот, не существует такого параметра, который не входил бы ни в один уровень защиты.

С каждым параметром безопасности $p_i \in P$ связано множество m_i, являющееся подмножеством множества M и включающее в себя параметр безопасности p_i. При этом параметр p_i не может входить во все политики безопасности, входящие во множество M, и, вместе с этим, параметр p_i не может не входить ни в одну из политик безопасности.

С каждой политикой безопасности $m_i \in M$ связано множество p_i, являющееся подмножеством множества P и включающее в себя параметры, входящие в m_i. При этом не существует такой политики безопасности, в которую бы не вошел ни один из параметров безопасности, и, вместе с тем, каждой политике безопасности соответствует свой набор параметров.

Состояние безопасности объекта определяется множеством Z, которое включает в себя множество реализованных угроз $T_\text{р}$ за некоторый интервал времени ΔT.

Элементы, включаемые в модель, назовем "актуальными", обладающими той или иной "значимостью" как мерой актуальности. Актуальность, по своей сути, — характеристика, зависящая от конкретной ситуации, и не имеет смысла, если не определены исходные условия, ограничения, не задан критерий, относительно которого можно судить об актуальности и оценивать значимость.

В качестве критерия r_z будем рассматривать сформулированную в явном виде и в содержательных терминах цель, которая ставится для защищаемого объекта для повышения его уровня безопасности относительно исходного. В самом общем виде целью является перевод системы из исходного состояния меньшей защищенности $z_0 \in Z$ в другое, желаемое состояние большей защищенности z^*. Цель реализуется СЗИ с использованием политик безопасности [7].

Структурно-функциональная схема модели представлена на рис. 3.

Роль условного элемента z, соответствующего индикатору состояния защищенности объекта в целом, как преобразователя, ограничивается функцией сумматора. Тогда на выходе z можно фиксировать результирующий поток f_z, интеграл от которого F_T по интервалу времени ΔT, является абсолютным показателем уязвимости объекта, измеряемой ущербом, наносимым ему за некоторое время T. Таким образом, задача СЗИ состоит в преобразовании результирующего потока угроз f_z в f_z^*, интегральная характеристика которого

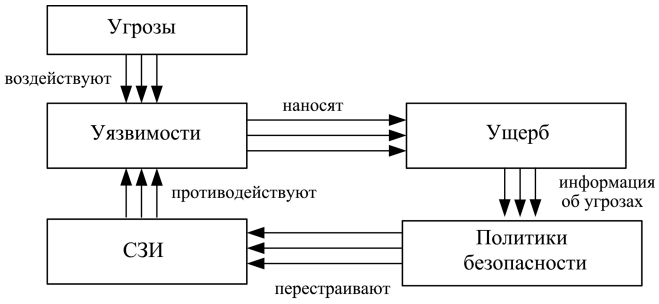


Рис. 3. Структурно-функциональная схема модели

должна уменьшаться до уровня F_T^*, не выше заданного в качестве цели F_T^0, что и будет означать переход объекта в более защищенное (менее уязвимое) состояние z^*. Тогда эффективность применения СЗИ измеряется следующим относительным показателем:

$$r_z = 1 - \frac{F_T^*}{F_T}, \quad r_z < 1,$$

а качество защиты как мера достижения потребительской цели — показателем

$$q_z = \frac{(F_T - F_T^*)}{(F_T - F_T^0)} = \frac{r_z}{r_z^0}, \quad r_z^0 > 0,$$

где r_z^0 — целевой, поддерживаемый с начального состояния, уровень защищенности.

Иногда удобно использовать инвариантные r_z показатели относительного, по сравнению с исходным состоянием, уровня защищенности и уровня уязвимости объекта соответственно:

$$s_z = \frac{F_T}{F_T^*} = (1 - r_z)^{-1}, \quad s_z > 0;$$

$$u_z = \frac{F_T^*}{F_T} = s_z^{-1} = 1 - r_z, \quad u_z > 0.$$

Свою задачу СЗИ решают путем осуществления воздействий на элементы множества V за счет изменения значений параметров множества P. Характер воздействий в содержательном смысле может быть самый разнообразный, но их результат для данной модели сводится к уменьшению числа успешно реализованных угроз.

Таким образом, в работе предложена модель определения состояния защищенности компьютерной сети путем формирования наборов параметров безопасности для каждого уровня защиты сети. Разработанная модель позволяет осуществлять динамический контроль над состоянием защищенности компьютерной сети и политикой безопасности за счет своевременного перестроения параметров политики безопасности. Данный подход предусматривает возможность формирования новых политик безопасности как на основе существующих, за счет возможности добавления новых параметров безопасности, так и без учета тех параметров, которые использованы в существующей модели.

Список литературы

1. **Доля А.** Проактивные технологии для борьбы с вирусами // Экспресс Электроника. 2006.
2. **Котенко И. В., Юсупов Р. М.** Перспективные направления исследований в области компьютерной безопасности // Защита информации. Инсайд. 2006. № 2. С. 46—57.
3. **Шаблоны** для контроля отдельных вопросов информационной безопасности. http://www.infosec.ru/
4. **Мак-Клар С., Скембрей Дж., Курц Дж.** Секреты хакеров. Безопасность сетей — готовые решения. М.: Вильямс, 2004.
5. **Оре О.** Теория графов. М.: Наука, 1980.
6. **Стратегия** защиты предприятия. http://www.trendmicro.com.
7. **Шишкин В. М.** Метамодель анализа, оценки и управления безопасностью // Проблемы управления информационной безопасностью: Сб. трудов Института системного анализа Российской академии наук. М.: Едиториал, 2002. С. 92—105.

WEB-ТЕХНОЛОГИИ

УДК 004.44/4

А. А. Чеснавский, аспирант,
Московский государственный инженерно-физический институт (ГУ)

Практическое применение алгоритма семантического анализа изменений в HTML-документах

Рассматривается алгоритм семантического отслеживания изменений (АСОИ), который позволяет выявить изменения данных в теле HTML-документа, а не изменений разметки документа. Особенностью данного алгоритма является то, что не требуется проводить предобработку документа и знать внутреннюю структуру HTML-страницы. АСОИ может быть использован в различных практических задачах, где требуется манипулировать данными, полученными с веб-сайтов. В качестве основных примеров можно привести семантический веб-клиппинг, кэширование страниц, получение RDF-представления HTML-страниц.

Ключевые слова: *семантический анализ изменений, веб-клиппинг, структура данных веб-страницы, HTML-документ.*

Введение

На сегодняшний день одним из базовых средств интеграции унаследованных веб-приложений является использование портальной платформы как единой точки доступа. Наиболее популярными платформами на сегодняшний день являются *Microsoft Office SharePoint Server 2007, IBM WebSphere Portal Server и Oracle Portal.*

Порталы, первоначально являясь средством единого места доступа к Web-ресурсам, сейчас, предоставляя дополнительные функции по организации совместной работы, делопроизводства, обеспечения безопасности, поиска, превратились в унифицированное информационное пространство для коллективной работы.

Полнофункциональное решение для портала должно предоставлять пользователям удобный доступ ко всему, что им необходимо для выполнения своих задач, вне зависимости от времени и места, а также при гарантии информационной безопасности.

Современные портальные решения, базирующиеся на Web-технологиях, реализованы в большинстве случаев на платформе J2EE (Java2 Enterprise Edition) в виде компонентов сервера приложений. Портальный сервер (Portal engine) представляет собой приложение, выполняющееся в среде сервера приложений, именно поэтому наиболее полнофункциональные портальные решения предлагаются компаниями, лидирующими в этой области.

Создание информационного наполнения портала осуществляется с помощью портлетов — компонентов сервера портала, обеспечивающих доступ к приложениям, Web-содержимому и другим информационным ресурсам и выполняющих функцию визуализации предоставляемых ими данных (рис. 1).

Каждый отдельный портлет разрабатывается, развертывается, управляется и отображается независимо от других. Администраторы и конечные пользователи могут создавать персонализированные страницы портала путем выбора и настройки, соответствующих портлетов. Режимы портлетов позволяют в зависимости от требуемой задачи отображать различные варианты пользовательского интерфейса.

В состав коммерческих портальных решений, как правило, включен обширный набор готовых к использованию портлетов, предназначенных для совместной работы пользователей, визуализации хранящихся в СУБД данных, выполнения XML-преобразований, а также для организации доступа к IBM Lotus, Microsoft Exchange, SAP/R3, Siebel и т. д.

Поскольку портлеты продолжают эволюционировать как новый стандарт для настольных приложений и средств интеграции, производители порталов прикладывают значительные усилия

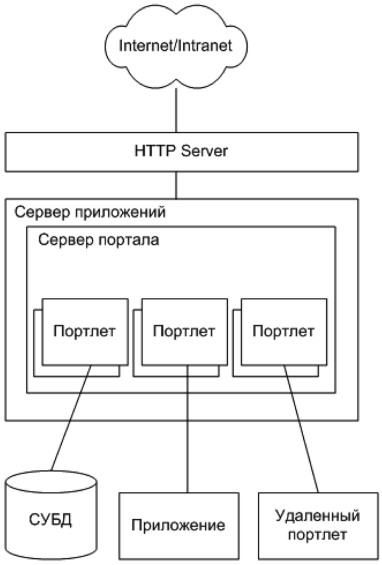


Рис. 1. Общая архитектура портального решения

для стандартизации прикладных программных интерфейсов, которые необходимы для взаимодействия между порталами и другими приложениями. Сообщество *Java Community Process* (JCP) и *Organization for the Advancement of Structured Information Standards* (OASIS) совместно работают над проблемой стандартизации технологий Java и XML, что обеспечит соединение порталов с разнородными приложениями.

В настоящее время корпоративные порталы активно используются для интеграции информационных систем в единое информационное пространство предприятия. Одним из самых востребованных механизмов интеграции является синдикация веб-контента. С точки зрения синдикации контента возможности портальных платформ можно разделить на две части: синдикация контента из XML-источников и синдикация контента из других источников. Наиболее востребованными XML-источниками являются RSS-ленты, удаленные портлеты и т. п. В качестве не-XML-источников можно указать HTML-страницы. Синдикация контента из первой группы практически во всех портальных платформах представлена в достаточном виде, в то время как синдикация контента из HTML-страниц (веб-клиппинг) может быть с трудом представлена на практике. Так, например, в *Microsoft SharePoint Server 2007* и *IBM WebSphere Portal Server* веб-клиппинг представлен только в виде помещения целевой страницы в выделенный портлет на странице (с помощью *iframe*). *Oracle Portal* предоставляет большую функциональность, позволяя помещать в портлет только фрагмент страницы, но во всех реализациях портальных платформ не реализован механизм семантического анализа структуры и синдикации значимых данных HTML-страниц.

В данной статье рассматривается практическое применение интеграции веб-контента в корпоративный портал на основе семантического анализа изменений веб-страниц.

Существующие алгоритмы поиска изменений

Как уже указывалось выше, существующие реализации веб-клиппинга имеют существенный недостаток — они не позволяют определить семантические изменения документов HTML, что связано с тем, что HTML — это во-первых, не просто файл, а полуструктурированный текст, а во-вторых, наряду с данными в нем содержатся и элементы представления данных (разметка данных). Все это существенно усложняет анализ изменений HTML-страниц.

На данный момент автору статьи неизвестны алгоритмы семантического анализа изменений HTML-страниц. Существующие алгоритмы ориентированы либо на анализ изменений в "плоских" документах, в XML-документах, либо на синтаксический анализ изменений в HTML-документах.

Так, одним из наиболее популярных инструментов для анализа изменений в "плоских" файлах является GNU-утилита **diff**. Эта программа выводит построчно изменения, сделанные в файле (для текстовых файлов). Работа **diff** основана на поиске наибольшей общей подпоследовательности (англ. *longest common subsequence*, LCS) [2]. В целом, задача нахождения наибольшей общей подпоследовательности является одной из классических задач информатики и применяется не только в таких утилитах как **diff**, но и в биоинформатике. Если вкратце описать суть алгоритма, то последовательность Z является общей подпоследовательностью последовательностей X и Y, если Z является подпоследовательностью как X, так и Y. Требуется для двух последовательностей X и Y найти общую подпоследовательность наибольшей длины. Очевидно, что данный алгоритм не подходит для анализа иерархических, а тем более HTML-документов. Конечно, иерархические документы можно сериализовать и затем применить к ним утилиту **diff**, но это будет неэффективно.

Вторую группу составляют алгоритмы, ориентированные на анализ изменений в иерархических документах, в частности в XML-файлах. Эти алгоритмы в большинстве случаев основываются на сравнении деревьев (благодаря тому, что иерархические документы представимы в древовидной форме). В 1979 г. Kuo-Chung Tai представил первый неэкспоненциальный алгоритм сравнения двух деревьев на основе расстояния редактирования [10]. До этого в 1977 г. Selkow [9] предложил довольно близкий к XML алгоритм преобразования деревьев — рекурсивный алгоритм поиска наибольшей общей подпоследовательности. Позже S. Chawathe [3—6] предложил два алгоритма — MMDiff и XMDiff (для основной и внешней памяти соответственно) для анализа изменений в упорядоченных деревьях, основанных на алгоритме Selkow.

Если рассматривать неупорядоченные деревья, то задача становится *NP*-сложной и необходимы дополнительные ограничения для сравнения двух деревьев. Так, можно выделить алгоритмы K. Zhang [12] и X-Diff [11], созданные для решения этой задачи. Еще одной утилитой, достойной внимания, является DeltaXML (по мнению ряда аналитиков – одна из лучших утилит для анализа изменений в XML-документах [7]). Эта утилита использует алгоритм, основанный на поиске наибольшей общей подпоследовательности и обладает линейной сложностью. Если говорить про анализ изменений на HTML-страницах, то практически единственной на сегодняшний день утилитой является HtmlDiff. HtmlDiff рассматривает HTML-документ как последовательность токенов, которые формируются на основе разметки и текста. В основе HtmlDiff лежит взвешенный алгоритм поиска наибольшей общей подпоследовательности. Результатом работы этой утилиты является синтаксический анализ отличия между двумя HTML-документами. На основе HtmlDiff создан ряд других утилит для анализа изменений: AT&T Internet Difference Engine, CS-HTMLDiff.

В целом, существует довольно ограниченное число утилит и соответствующих алгоритмов, подходящих для анализа изменений в иерархических документах. Если же рассматривать класс алгоритмов для анализа изменений в HTML-документах, то все известные автору алгоритмы ориентированы на синтаксический анализ изменений, что имеет невысокую применимость в более общей задаче веб-клиппинга ввиду того, что необходимо прежде всего анализировать значимые изменения на веб-страницах.

Рассмотрим пример HTML-документа [8] (рис. 2) на основе некоторой статистической информации о плодовых мушках.

```
<TABLE BORDER>
<CAPTION>A test table with merged cells. <CREDIT> (T. Berners Lee/WWWC,1995.) </CREDIT></CAPTION>
<TR><TH ROWSPAN = 2><TH COLSPAN = 2> Average<
<th rowspan = 2> Red <br> eyes
<TR><TH> height <TH> weght
<TR><TH ALIGN = left> males <TD> 1.9 <TD> 0.003<
<td> 40 %
<TR><TH ALIGN = left> females <TD> 1.7 <TD> 0.002<
<td> 43 %
</TABLE>
```

Предположим, что значение 40 % в последней колонке изменяется на 50 %, это означает, что значение процента красных глаз у особей мужского пола изменилось с 40 на 50 %. Для того, чтобы технически осуществить данное изменение, нужно в HTML-коде в пятой строке 40 % заменить на

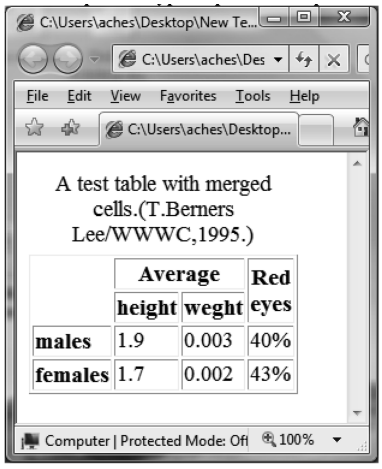


Рис. 2. Пример таблицы

50 %. Как уже говорилось выше, при синтаксическом анализе изменений сообщение об изменении в пятой строке было совсем неинформативным. Даже структурированное описание изменения в HTML-коде, основанное на HTML-грамматике, такое как "TABLE. TR. TD.40 %" изменено на "TABLE. TR. TD.50 %" не является достаточно информативным, и с его помощью тяжело отследить реальное изменение процента красных глаз у особей мужского пола.

Подобные проблемы могут быть успешно разрешены с помощью алгоритма семантического отслеживания изменений (АСОИ).

Алгоритм семантического отслеживания изменений

АСОИ отслеживает изменения, которые мы назовем семантическими изменениями, в HTML-документах в рамках иерархии данных, в отличие от иерархии разметки HMTL-документа. Используя описанный выше пример, АСОИ определил бы изменения как "Males.'Red eyes' изменен с 40 % на 50 %". Особенность АСОИ состоит в адаптации понятия семантических изменений для отслеживания изменений в HTML-документах. В противоположность другим хорошо спроектированным полуструктурированным данным или XML-документам путь между корневой вершиной и листовым узлом в дереве анализа HTML-документа (например, TABLE. TR. TD.40 %) не обязательно описывает значение самого узла, так как HTML определяет к тому же и представление данных. В отличие от HTML, в хорошо структурированных документах путь между вершиной и листовым узлом в основном информативен и значим. Более того, XML требует, чтобы каждый элемент был закрытым, в то время как закрывающие тэги у некоторых HTML-элементов могут отсутствовать или быть необязательными, что приводит к сложностям в разборе (*parsing*) HTML-документов. В результате, ввиду этих отличий получение информации из HTML-документов требует дополнительных знаний о внутренней структуре или предобработки исходных документов [1] (что в реальной практике может быть недоступно), а это не нужно для работы АСОИ.

АСОИ состоит из следующих шагов:

- конструирование семантической иерархии (дерева) HTML-документа;
- анализ изменений семантической иерархии.

Структурно HTML-документ состоит из одной или более логических секций, которые:

- находятся друг относительно друга на одном уровне, например Section 1, Section 2, и т. д.;
- одна секция структурно включает другую, например Section 1 и Section 1.2;
- две секции не находятся на одном уровне, и одна из них не включает другую, например Section 1.3 и Section 4.

Прежде всего, необходимо определить семантическую иерархию секций в HTML-документе, используя различные HTML-тэги. Как уже было сказано выше, HTML был создан не только для определения, но и для отображения данных и, следовательно, большинство HTML-документов не способствуют организации компонентов HTML в секции или блоки согласно иерархии. Таким образом, в первую очередь необходимо идентифицировать, какие HTML-тэги могут быть использованы для

Группы HTML-тегов

HTML-тэги			Тип 1	Тип 2
Head			TITLE, META	ISINDEX, BASE, LINK, SCRIPT, STYLE, META
Body	Заголовки		H1, H2, H3, H4, H5, H6	
	Блоки		P, CENTER, BLOCKQUOTE, PRE, DIR, MENU, DL, DT, DD, UL, OL, LI, TABLE, CAPTION, THEAD, TBODY, TR, TH, TD	ISINDEX, HR, DIV
	Текст	Шрифт		TT, I, B, U, STRIKE, BIG, SMALL, SUB, SUP
		Фраза		EM, STRONG, DFN, CODE, SAMP, KBD, VAR, CITE
		Специальный	IMG	A, APPLET, FONT, BASEFONT, BR, SCRIPT, MAP
		Форма		FORM, INPUT, SELECT, TEXTAREA
	Адрес		ADRESS	

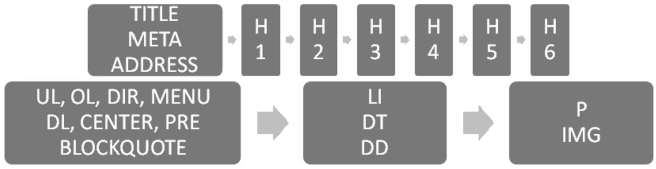


Рис. 3. Порядок предшествования нетабличных элементов (тип 1)

конструирования иерархической структуры HTML-документов (тип 1), а какие служат для представления данных (тип 2). Список тегов с разделением по типам можно найти в таблице.

Конструирование семантической иерархии для нетабличных данных состоит из двух шагов. На первом шаге все тэги типа 2 удаляются из исходного HTML документа. Отметим, что удаление тэгов типа 2 может привести к конкатенации #PCDATA. Например, <LI><I>text 1</I>text 2</LI> приводит к <LI>text 1 text 2</LI> после удаления тэга <I>. На втором шаге семантическая иерархия конструируется на основе предшествования нетабличных HTML-тэгов так, как это изображено на рис. 3. Предшествование между двумя HTML-элементами A и B, обозначаемое $A \gg B$, показывает, что данные, содержащиеся в A, выше в соответствующей иерархии, чем данные, содержащиеся в B.

Рассмотрим табличные элементы HTML. Среди табличных элементов TR определяет число строк, тогда как TH и TD определяют число столбцов в HTML-таблице. Элемент TH используется для задания одного или более заголовков. Элемент TD используется для внесения данных в ячейки таблицы. В дальнейшем будем называть данные в элементах TD табличными данными, в отличие от данных, содержащихся в элементах TH, которые будем называть заголовками.

Типовая HTML-таблица имеет как минимум один столбец-заголовок в верхней части таблицы и как минимум одну строку-заголовок в левой части. Такой тип таблиц мы назовем строчно-столбцовым. Другой тип таблицы содержит как минимум один столбец-заголовок (одну строку-заголовок) и называется в этом случае столбцовым (строчным соответственно) типом таблицы. Заголовки в строчных и столбцовых таблицах задают схему таблицы. Для любых таблиц, которые не имеют элементов TH, в ходе анализа было выявлено, что первая строка или столбец обычно используется как заголовок.

Среди табличных элементов два атрибута TH и TD, ROWSPAN и COLSPAN играют существенную роль в определении иерархии HTML-таблиц. Для иллюстрации рассмотрим пример, приведенный выше (см. рис. 2). В данном примере наблюдается различное число строк и столбцов, что затрудняет процесс корреляции строк и столбцов. Когда TH или TD включает ROWSPAN = "n" (COLSPAN = "n" соответственно) связанная ячейка распространяется на N столбцов вниз (N строк вправо соответственно).

Для определения семантической иерархии (SH), расширяющей синтаксическое дерево любой HTML-таблицы T, мы в первую очередь определяем иерархические зависимости данных в T. Как только иерархические зависимости определены, SH содержит только данные, и все тэги исключены из T.

Семантическая иерархия HTML-таблицы определяется согласно нотации псевдотаблицы, так как свойства псевдотаблицы легки для восприятия. Псевдотаблица может рассматриваться как особый тип HTML-таблицы и может быть использована для выражения строчно-столбцовых, строчных и столбцовых таблиц. Общая схема построения семантической иерархии — это в первую очередь отображение таблицы T на псевдотаблицу и затем получение из нее семантической иерархии [13].

Практическое применение АСОИ

В результате работы АСОИ применительно к определенной веб-странице мы получаем семантическую иерархию, которая имеет следующие характеристики:

- отсутствие информации о форматировании;
- устойчивость относительно некардинальных изменений в форматировании HTML-страницы.

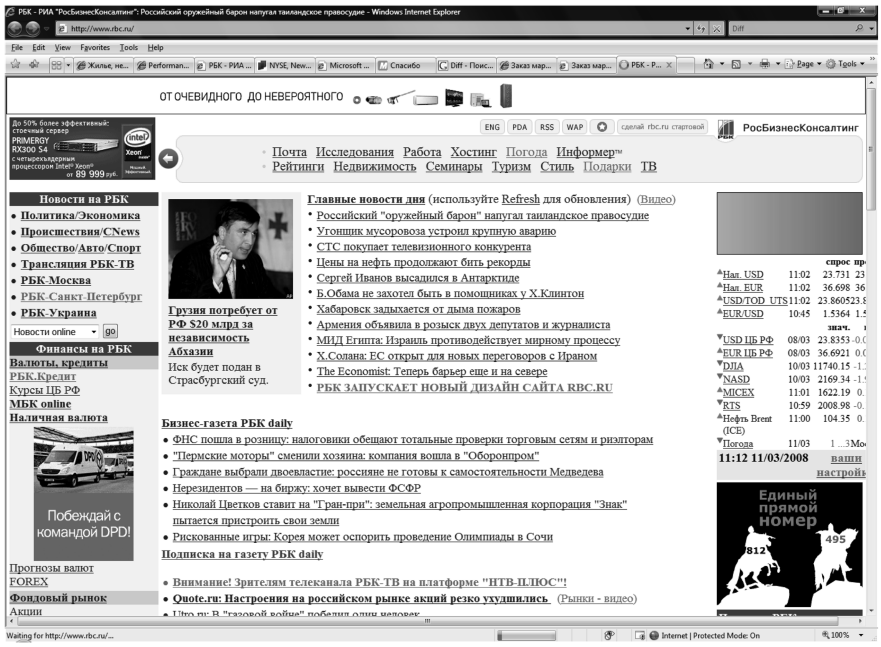


Рис. 4. Скриншот сайта РБК

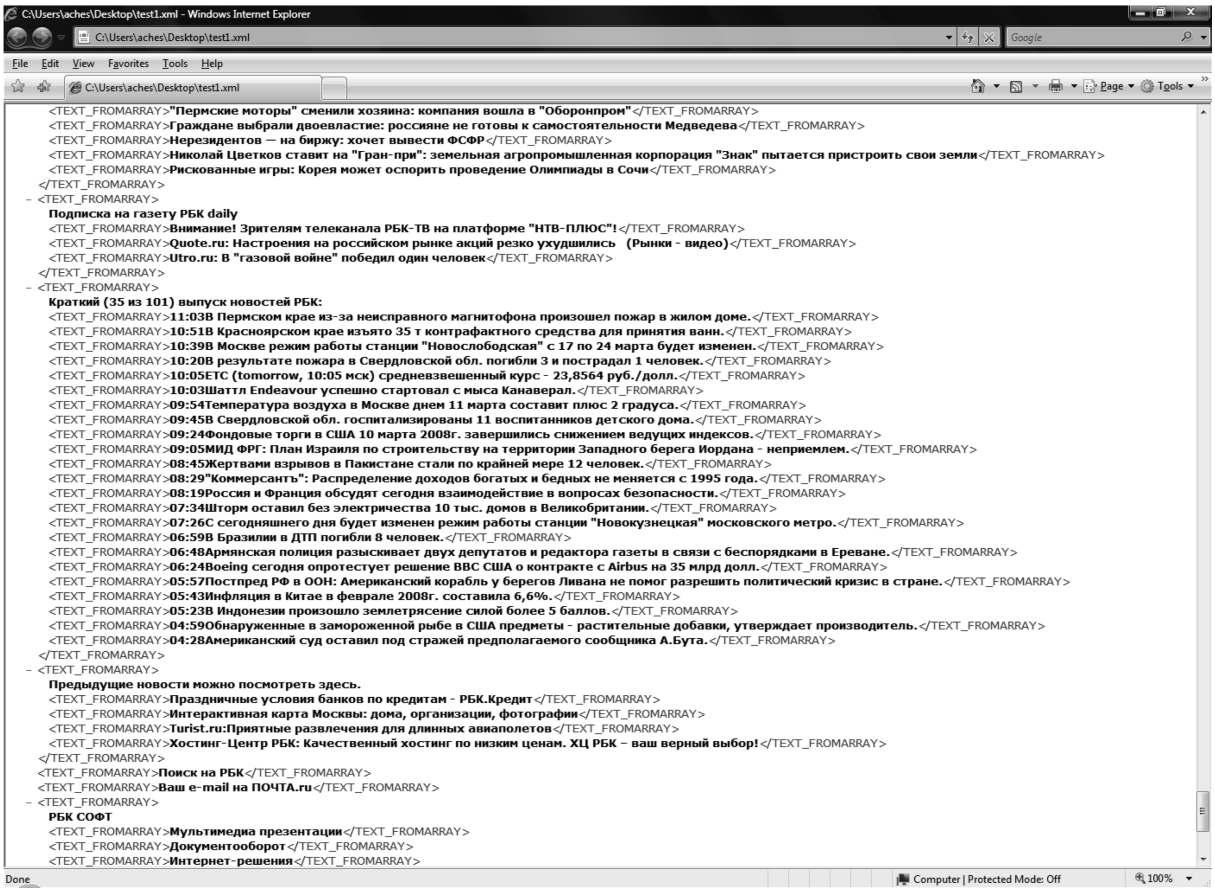


Рис. 5. Семантическая иерархия сайта РБК

Пример получения семантической иерархии для популярного аналитического сайта РБК (www.rbc.ru) (рис. 4) представлен на рис. 5.

Следует отметить, что, получив семантическую иерархию веб-сайта в виде XML-документа, можно реализовать следующие варианты практического применения АСОИ:

- оптимизация кэширования в алгоритмах веб-клиппинга;
- публикация семантической структуры сайта в формате RDF;
- анализ изменений отдельных узлов семантической иерархии.

Рассмотрим более подробно каждый из этих сценариев. Как уже говорилось выше, в настоящее время при разработке промышленных портальных решений одной из востребованных, но слабореализуемых на практике задач является синдикация HTML-контента в единое информационное пространство на базе портала, т. е. публикация содержимого существующей веб-страницы в отдельный портлет в рамках корпоративного портала. Основная задача данного портлета — это предоставление по запросу пользователя содержимого конкретной HTML-страницы. Однако это только кажущаяся простота. Пользователями данного портлета могут быть тысячи людей с различной динамикой обращения к странице. В простейшем варианте реализации подобный портлет при каждом запросе пользователя будет формировать запрос от сервера портала к целевой странице, что в совокупности сгенерирует очень высокий трафик и нагрузку на систему. Гораздо эффективнее организовать кэширование запрашиваемых страниц. Однако основная проблема отображения закэшированных данных — это обеспечение актуальности (свежести) данных. В подавляющем большинстве случаев пользователей портлетов с веб-клиппингом интересует семантика внешней страницы, а не форматирование. И изменения в кэше должны происходить только в случае изменения семантической структуры страницы. Таким образом, АСОИ может быть использован как основа для анализа необходимости обновления кэша.

В последние несколько лет в среде ИТ-профессионалов все активнее обсуждаются такие концепции, как "семантическая сеть", "Web 2.0" и т. п. Семантическая сеть (также известная как семантический веб) — это расширение существующей сети (Интернета), в которой информация снабжена смыслом, позволяющим человеку и компьютеру успешно взаимодействовать. Для иллюстрации различия между WWW и семантической сетью можно

привести следующий пример: допустим необходимо провести поиск по ключевым словам в Интернете. В качестве результата будут получены ссылки, многие из которых окажутся не совсем релевантными. Это связано с тем, что информация в WWW хранится по большей части в виде HTML-страниц, которые не предназначены для автоматической обработки и формирования формальной семантической структуры. Сеть нового поколения должна обеспечивать возможность автоматизированной интерпретации и обработки информации, семантической интероперабельности информационных ресурсов. Необходимость решения указанных задач вызвала потребность в таких средствах формального описания семантики XML-данных, которые бы позволяли анализировать и обрабатывать их с помощью программного обеспечения. Консорциум W3C предложил многоуровневую архитектуру для семантической сети (рис. 6, см. вторую сторону обложки).

В основе этой структуры лежит стандарт RDF (*Resource Description Framework*). RDF представляет собой простой способ описания экземплярных данных в формате субъект—отношение—объект, в котором в качестве любого элемента этой тройки используются только идентификаторы ресурсов. Существует стандартизованное отображение этих троек на XML-документы предопределенной структуры. Благодаря тому, что результатом работы АСОИ является формальная семантическая иерархия, которая может быть преобразована не просто в XML, но и в RDF, появляется возможность автоматического получения семантики веб-страниц в формате RDF, что в дальнейшем позволит манипулировать данными веб-страниц с помощью соответствующих инструментов.

И наконец, рассмотрим применение АСОИ для автоматического анализа изменений отдельных элементов семантической иерархии HTML-страницы. Рассмотрим пример: трейдер формирует портфель из нескольких финансовых инструментов. Текущие котировки берутся со специализированного веб-сайта, содержащего данные по всем эмитентам. С помощью связки АСОИ (для получения семантической иерархии) и портала (публикация значения выделенного элемента) трейдер может избежать постоянного мониторинга массива данных по котировкам и получать на свою персональную страницу только данные по нужным ему эмитентам. Более того, так как семантическая иерархия представляет собой XML-документ, то к отслеживаемым элементам иерархии можно применять выражения Xpath. Так, в частности, можно выставить пороговые значения для элемента и проводить какие-либо действия при выходе за эти границы (например, информировать пользователя о достижении критических значений).

Рассмотрим использование данной схемы на практике. В качестве источника данных будем использовать данные по котировкам акций на ФБ ММВБ на сайте QUOTE. RU (РБК) (рис. 7, см. вторую сторону обложки). В исходной HTML-странице находятся данные по более чем 300 эмитентам с несколькими значениями котировок для каждого из них, т. е. порядка 3000 значений, среди которых трейдер вынужден искать интересующую его информацию.

В результате работы АСОИ мы получим семантическую иерархию (см. рис. 5) — XML-документ, который дальше может быть использован для получения информации по котировкам отдельного эмитента.

Ключевой задачей, которую необходимо решить при мониторинге отдельного элемента семантической иерархии — это позиционирование элементов в семантической иерархии, т. е. определение координат каждого элемента. Наиболее подходящее решение этой задачи — использование Xpath-пути к элементу в качестве идентификатора. Таким образом, применением специального XSLT-преобразования к исходному XML-документу можно получить семантическую иерархию с помеченными элементами. Преимущество такого подхода заключается в том, что идентификатор без каких-либо изменений можно использовать в Xpath-выражении для поиска элемента.

Последним шагом в задаче мониторинга элементов семантической иерархии является публикация значения элемента, например, на персональной странице пользователя на портале. Пример использования АСОИ для мониторинга значений акций "Полюс Золото" в портале на основе Microsoft Office SharePoint Server 2007 представлен на рис. 8 (см. вторую сторону обложки).

Заключение

В данной статье предложен алгоритм семантического отслеживания изменений (АСОИ), который позволяет выявить изменения данных в теле HTML-документа, а не изменений разметки документа. Особенность данного алгоритма в том, что не требуется проводить предобработку документа и знать внутреннюю структуру HTML-страницы. АСОИ может быть использован в различных практических задачах, где требуется манипулировать данными, полученными с веб-сайтов. В качестве основных примеров можно привести семантический веб-клиппинг, кэширование страниц, получение RDF-представления HTML-страниц.

Список литературы

1. **Atzeni P., Mecca G.** Cut and Paste / Paolo Atzeni, Giansalvatore Mecca // Proc. of the sixteenth ACM SIGACT-SIGMOD-

SIGART symposium on Principles of database systems. 1997. C. 144—153.

2. **Bergroth L., Hakonen H.** A Survey of Longest Common Subsequence Algorithms [Текст] / L. Bergroth, H. Hakonen // Proc. of the Seventh International Symposium on String Processing Information Retrieval (SPIRE'00). 2005. C. 39.

3. **Chawathe S.** Comparing Hierarchical Data in External Memory [Текст] / Sudarshan S. Chawathe // Proc. of the 25th International Conference on Very Large Data Bases. 1999. — C. 90—101.

4. **Chawathe S., Abiteboul S., Widom J.** Representing and querying changes in semistructured data [Текст] / Sudarshan S. Chawathe, Serge Abiteboul, Jennifer Widom // Proc. of the International Conference on Data Engineering. 1998. — C. 4—13.

5. **Chawathe S., Garcia-Molina H.** Meaningful Change Detection in Structured Data [Текст] / Sudarshan S. Chawathe, Hector Garcia-Molina // Proceedings of the ACM SIGMOD International Conference on Management of Data SIGMOD. 1997. — C. 26—37.

6. **Chawathe S., Rajaraman A., Garcia-Molina H., Widom J.** Change detection in hierarchically structured information [Текст] / Sudarshan S. Chawathe, Anand Rajaraman, Hector Garcia-Molina, Jennifer Widom // Proc. of the ACM SIGMOD International Conference on Management of Data. 1996. — № 25 (2). — C. 493—504.

7. **Hinze A., Evans R.** Keeping Track of the Semantic Web: Personalized Event Notification [Текст] / Annika Hinze, Reuben Evans // On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. — 2006. — № 4275. — C. 661—678.

8. **HTML 4.0** Specification [Электронный ресурс] / Dave Raggett, Arnaud Le Hors, Ian Jacobs; W3C. — Электрон. дан. — [USA], 1998. — Режим доступа: http://www.w3.org/TR/1998/REC-html40-19980424/, свободный — Загл. с экрана. — Яз. англ.

9. **Selkow S.** The tree-to-tree editing problem [Текст] / Stanley M. Selkow // Information Processing Letters. — 1977. C. 184—186.

10. **Tai K.** The tree-to-tree correction problem [Текст] / Kuo-Chung Tai // Journal of the ACM. — 1979. — № 26 (3). — C. 422—433.

11. **Wang Y., Dewitt P., Cai J.** X-Diff: An Effective Change Detection Algorithm for XML Documents [Текст] / Yuan Wang, David J. DeWitt, Jin-Yi Cai // Proceedings of the 19th International Conference on Data Engineering. — 2003. — C. 519—530.

12. **Zhang K.** A Constrained Edit Distance Between Unordered Labeled Trees [Текст] / Kaizhong Zhang // Algorithmica. — 1996. — № 15 (3). — C. 205—222.

13. **Чеснавский А.** Семантическое отслеживание изменений на веб-сайтах [Текст] / Чеснавский А. // Управление большими системами. — 2007. — Вып. 19. — C. 134—153.

УДК 004.056.5

Д. Л. Жусов, адъюнкт,
В. В. Комашинский, канд. техн. наук, зам. нач. каф.,
Академия ФСО России, г. Орел

Варианты реализации модуля фильтрации потока запросов к Web-серверу

Предложены варианты реализации модуля фильтрации потока запросов к Web-серверу с динамически формируемыми страницами, позволяющие повысить его защищенность от компьютерных атак подмены контента.

Ключевые слова: *модуль фильтрации, поток запросов, Web-сервер, защита, компьютерные атаки.*

Введение

В настоящее время в России большое внимание уделяется вопросам совершенствования системы государственного управления, повышения качества предоставления информации населению и организациям [1, 2]. В системе информационного обеспечения органов государственной власти (ОГВ) важное место занимают Web-серверы, предназначенные для предоставления информации и обеспечения межведомственного взаимодействия и взаимодействия ОГВ с населением и организациями. При этом на официальных Web-сайтах информация открыта для пользователей сети Интернет и не содержит конфиденциальных сведений.

Сегодня Web-серверы предусматривают использование статического метода формирования Web-страниц. Данный подход позволяет достаточно легко реализовать процедуры их защиты. Однако его недостатками являются низкая информативность Web-сайтов, трудоемкость ввода новой информации в структуру сайта и невозможность организации поиска информации по запросам пользователей, которые устраняются при переходе к динамическому методу формирования Web-страниц. На практике это приводит к тому, что против Web-сайтов могут быть организованы специальные компьютерные атаки, направленные на подмену информации на Web-страницах (атаки подмены контента) [3]. В связи с этим актуальной является задача обеспечения безопасного функционирования Web-серверов в условиях компьютерных атак подмены контента и практической реализации соответствующих технических решений.

Результаты исследования

Защита Web-серверов от компьютерных атак может быть реализована процедурой фильтрации потока запросов, основанной либо на методе обнаружения сигнатур, либо на методе обнаружения аномалий. Однако данные методы обладают взаимообратными достоинствами и недостатками.

Недостатки метода обнаружения сигнатур:

- принципиальная невозможность обнаружения новых атак и модификаций существующих;
- необходимость разработки и постоянного пополнения базы данных сигнатур.

Недостатки метода обнаружения аномалий:

- большое число ложных срабатываний;
- вероятность пропуска маскированных атак.

В связи с этим при разработке модуля фильтрации потока запросов к Web-серверу можно пред-

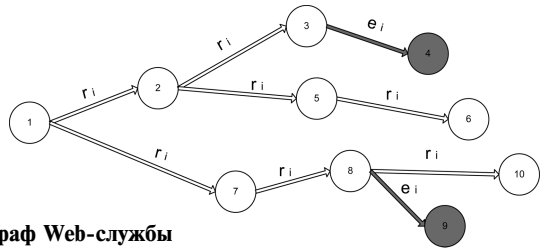


Рис. 1. Орграф Web-службы

положить, что он должен представлять собой комплексное представление существующих методов, при этом недостатки одного должны компенсироваться преимуществами другого.

Представим Web-службу как орграф $G = \{V, E\}$, где V — множество вершин графа; E — множество ребер (рис. 1). Вершина орграфа (состояние сервера) будем характеризовать результат выполнения запроса по методам доступа *GET* или *POST*, что, в свою очередь, характеризуется запрашиваемым файлом или Web-приложением с конкретными параметрами. Пусть орграф G — взвешенный орграф, функция $w(*)$ — функция веса, а W — множество всех запросов к серверу.

Подмножество R множества W ($R \subseteq W$) будем называть разрешенным, если $w(r_i) \in R$ запрос к Web-серверу не привел к нарушению безопасности системы.

В силу того, что любое состояние сервера однозначно определяется запросом к нему, и в орграфе нет смежных ребер, следовательно, он является деревом. Тогда дерево $G = \{S, V, R\}$ является моделью безопасной системы. Запрос e будет идентифицирован как возможная компьютерная атака или принят за ошибочный, если он порождает собой дерево $E \notin G$.

Первый вариант реализации модуля демонстрирует принцип построения графа и реализует функции построения безопасного дерева, проверки принадлежности запроса к множеству безопасных, поддержания актуальности безопасного дерева.

Множество R однозначно определяется статистическими и динамически сгенерированными ссылками. При разработке авторы исходили из предположения, что Web-сервер изначально, до начала эксплуатации, не может содержать заведомо атакующих запросов, и они создаются искусственно хакером. Следовательно, множество R можно сформировать на основе предварительного анализа ссылок на сайте. Если поступивший запрос не принадлежит безопасному множеству R, то он не будет нести полезной функциональности, т. е. состояния V не будут принадлежать дереву безопасной системы G, и его можно считать компьютерной атакой.

Данный вариант модуля фильтрации потока запросов имеет следующие преимущества: отсутствие необходимости хранения всех сигнатур атак, являющихся бесконечным множеством, и постоянного их обновления; быстрота проверки. Он реализован в программном модуле WIDS (рис. 2), который представляет собой прокси-сервер, работающий на сеансовом уровне.

Для функционирования прокси-сервера необходимо указать адрес Web-сервера и запустить систему. На предварительном этапе формируется безопасное дерево запросов, которое строится по принципу сканирования сервера и выявления ссылок перехода, определения параметров и допустимых значений этих параметров. В дальнейшем, на этапе функционирования сервера, поступивший запрос проверяется на соответствие дереву безопасных запросов.

Однако данный вариант имеет следующие недостатки: увеличение объема безопасного дерева при защите ресурсоемких Web-серверов; механизм составления безопасного дерева на практике не обеспечивает требуемой полноты вследствие динамичности Web-служб.

В связи с этим в рамках исследования был разработан модифицированный вариант модуля, схема функционирования которого представлена на рис. 3.

Модуль фильтрации потока запросов CorePlex реализован на Web-сервере Apache 2.2.2 и основан на применении криптографических средств защиты информации, где в качестве ключевой функции используется ГОСТ Р34.11—94. Принцип работы модуля заключается в том, что для каждой ссылки вычисляется значение хеш-функции, зависящее от ключа, и отсылается браузеру, при приеме сервером запроса это значение сверяется с правильным, хранящимся на сервере, в результате чего устанавливается подлинность запроса.

Модуль CorePlex состоит: из входного фильтра coreplin.so, работающего на стеке фильтров

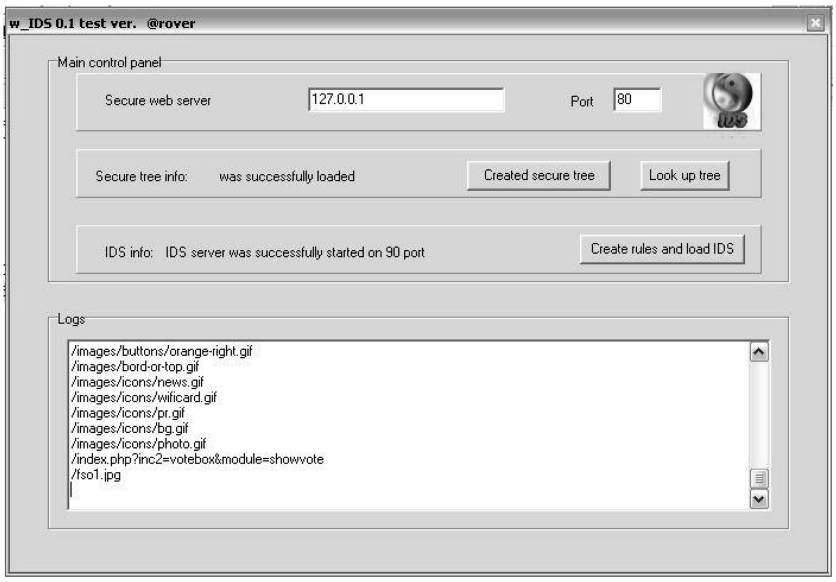


Рис. 2. Интерфейс модуля фильтрации потока запросов к Web-серверу WIDS

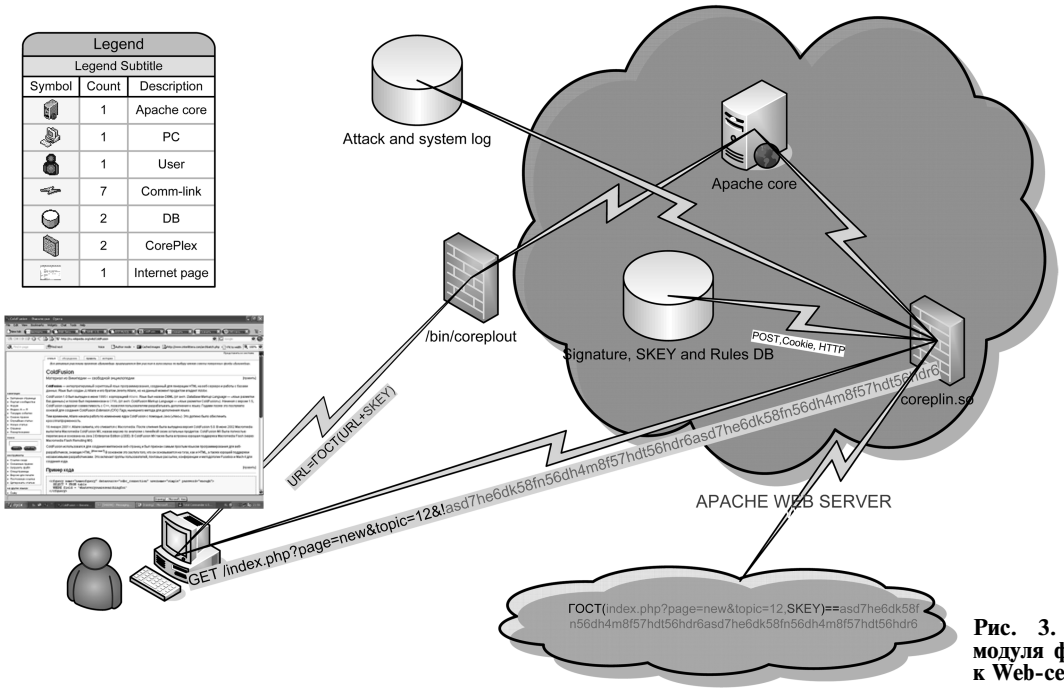


Рис. 3. Схема функционирования модуля фильтрации потока запросов к Web-серверу CorePlex

Apache; выходного фильтра coreplout, работающего через фильтр-посредник Apache; базы данных правил аномалий; сигнатур; ключевой информации и файлов журналов регистрации событий.

Функционирование модуля фильтрации потока запросов к Web-серверу может быть представлено следующими процедурами:

1-я процедура: пользователь посылает запрос на главную страницу сайта, данный запрос и все запросы, которые не содержат значений параметров на корректность, не проверяются.

2-я процедура: Web-сервер генерирует ответ и передает сгенерированную страницу на выходной фильтр, который выполняет операцию $\forall URL: URL = URL + H(URL + SKEY)$, где URL — запрос к Web-серверу; H(*) — функция хеширования, вычисляемая по ГОСТ Р34.11—94; $SKEY$ — секретный ключ (хранится на сервере и периодически изменяется). Данная контрольная сумма (значение хеш-функции) добавляется в кеш фильтра и пересчитывается только при изменении URL.

3-я процедура: обработанная страница отправляется клиенту.

4-я процедура: пользователь выбирает запрос с параметрами и посылает запрос серверу, данный запрос содержит проверочную сумму. Запрос поступает на входной фильтр, выполняется операция, аналогичная 2-й процедуре, и результат сравнивается с проверочной суммой (значением хеш-функции), поступившей в запросе. Если проверяемые значения совпадают, то запрос считается безопасным. Все остальные запросы, не содержащие параметры, проверяются сигнатурным способом.

Разработанный вариант модуля фильтрации потока запросов к Web-серверу имеет следующие преимущества:

- реализована политика использования безопасного дерева запросов и динамическая генерация контрольных сумм;
- независимость от платформы сервера;
- прозрачность для Web-приложений.

Заключение

В результате исследования разработаны варианты модуля фильтрации потока запросов к Web-серверу (WIDS и CorePlex), их программные реализации, позволяющие повысить защищенность Web-серверов с динамически формируемыми страницами от компьютерных атак подмены контента, а по представленным материалам получен патент на полезную модель [4].

Список литературы

1. **Концепция** использования информационных технологий в деятельности федеральных органов государственной власти до 2010 года // Распоряжение Правительства РФ от 27 сентября 2004 г. № 1244-р.

2. **Федеральная** целевая программа "Электронная Россия" (2002—2010 годы) // Постановление Правительства РФ от 28 января 2002 г. № 65.

3. **Синев С. Г., Иванов В. А., Комашинский В. В., Жусов Д. Л.** Способы и приемы компьютерных атак на Web-серверы с динамически формируемыми страницами // Вестник компьютерных и информационных технологий. 2007. № 3. С. 36—39.

4. **Пат. 70016.** Российская Федерация, МПК G 06 F 12/14. Устройство обнаружения компьютерных атак на информационные Web-ресурсы / Комашинский В. В., Васинев Д. А., Жусов Д. Л., Смыков Г. Г., Грызунов В. В.; заявитель и патентообладатель Государственное образовательное учреждение высшего профессионального образования Академия Федеральной службы охраны Российской Федерации (Академия ФСО России) — № 2007114605/22; заявл. 17.04.2007; опубл. 10.01.2008. Бюл. № 1. — 2 с.

УДК 004.657

С. С. Горелов, мл. науч. сотр.,
Научно-исследовательский институт
механики МГУ им. М. В. Ломоносова

Модели и алгоритмы для систем поиска в наборах документов

Рассматривается подход, позволяющий эффективно индексировать базы данных, представляющие собой наборы документов. Предложен алгоритм построения индекса и приведена оценка его сложности. Описанный подход позволяет применять оптимальные индексы для широкого класса задач, в частности, для случаев поиска информации в реляционных базах данных, в полуструктурированных базах OEM-документов, а также для поиска в наборах XML-документов.

Ключевые слова: *полуструктурированные базы данных; алгоритмы индексирования данных, вероятностная оценка эффективности поиска.*

Введение

На современном этапе развития информационных технологий сформировалась устойчивая тенденция достаточно быстрого роста размеров баз данных, используемых в тех или иных практически значимых приложениях. По этой причине актуальными являются исследования подходов к решению задач поиска и навигации по базам данных. В зависимости от различий в типах документов содержание процесса поиска может существенно различаться. В настоящей публикации предпринимается попытка с наиболее общих позиций подойти к решению задачи поиска в базах данных путем разработки универсальных методов оптимизации этого процесса.

Предложена теоретическая модель, оперирующая понятиями документа, схемы, запроса, базы данных и индекса. В рамках этой модели описаны алгоритмы поиска и индексирования.

В ходе реализации изложенного подхода возникает задача формулирования критериев сравнения индексов между собой с целью научиться строить оптимальные индексы. Вопрос оптимальности индексов рассматривается в настоящей работе с точки зрения средней стоимости вычислений при поиске в базах данных, аналогично тому, как это сделано в [1]. Такой подход заключается в формализации требования о том, чтобы запросы вычислялись за минимальное время. Запросы принимаются как случайные события, а время их выполнения рассматривается как математическое ожидание стоимости вычислений при поиске по всевозможным запросам. Для предложенных алгоритмов приведены оценки их сложности.

На основе введенной теоретической модели предложена архитектура системы, позволяющая осуществлять поиск и построение индекса для различных моделей документов, запросов и схем.

1. Формальная модель поиска и индексирования

В качестве основных функций, обеспечивающих взаимодействие с поисковой системой, отметим следующие две:

- построение индекса по базе данных;
- поиск в базе данных по запросу с использованием индекса.

Программная реализация отмеченных функций в контексте целей настоящей работы должна опираться на теоретическую модель поисковой системы, для которой будут описаны основные понятия и алгоритмы, реализующие заявленные функции. Формализуем основные понятия и опишем алгоритмы, соответствующие процессам поиска и индексирования в базах данных.

1.1. Поиск

В качестве базовых понятий, используемых для реализации алгоритмов поиска, рассмотрим: документ D — элемент заранее заданного множества $\mathcal{D}$; базу данных DB — конечное множество документов, $DB = \{D\}$; запрос — некоторый объект Q, являющийся элементом множества $\mathcal{Q}$.

Отметим, что при таком определении документ и запрос никак не связаны. Для того чтобы задать связь между ними, введем отображение Qd: $\mathcal{D} \times \mathcal{Q} \rightarrow \rightarrow \{0, 1\}$ (функцию поиска по документам).

Если для документа D и запроса Q верно равенство $Qs(D, Q) = 1$, то этот факт означает, что документ D соответствует запросу Q. Соответственно, равенство $Qs(D, Q) = 0$ означает обратное.

Для решения задачи сокращения времени поиска будем использовать индексы, представляющие собой деревья. Будем полагать, что в вершинах индекса (соответствующих деревьях) стоят некоторые объекты, называемые схемами, которые задают классы документов. Родительская схема при этом задает более общий класс документов, чем дочерняя.

В частных случаях применения излагаемой далее теории схемы будут задаваться конструктивно. Обозначая какую-либо схему как S, будем полагать,

что каждая схема S задает множество документов (соответствующих S), которое обозначим $[S]$. Множество всех схем обозначим как $\mathcal{S}$. Схема S_1 называется более общей, чем S_2, если $[S_2] \subseteq [S_1]$. Будем обозначать это отношение $S_1 \geqslant S_2$.

Индексом (*иерархией схем*) I для базы данных D назовем дерево, вершинам которого сопоставлены схемы, обладающие следующими свойствами:

- каждая схема является более общей, чем любая из ее дочерних схем;
- для каждого документа D из базы данных DB существует хотя бы одна схема индекса, которой он соответствует;
- для каждой листовой схемы должен существовать хотя бы один документ базы, который ей соответствует.

Для построения индекса в дальнейшем будет использовано понятие тривиальной схемы — S_0. Такая схема является обобщением всех схем множества $\mathcal{S}$ и ей соответствуют все документы $\mathcal{D}$.

Сокращение времени поиска будет основываться на том, что существует функция, позволяющая определить по запросу и схеме, что в документах, соответствующих схеме, найдено ничего не будет. Такую функцию будем называть функцией поиска по схеме и обозначать Qs: $\mathcal{S} \times \mathcal{Q} \to \{0, 1\}$. При этом $Qs(S, Q) = 1$, если $\exists D \in S : Qd(D, S) = 1$, и $Qs(S, Q) = 0$, если $\forall D \in S$: $Qd(D, S) = 0$.

Сформулируем и опишем алгоритм, который позволит сократить пространство поиска с помощью иерархии схем.

Алгоритм 1.1. ***Усечение пространства поиска.***

На каждом шаге алгоритма для рассматриваемой вершины S индекса I (для первого шага S — это корневая схема) проверяем условие Qs(S, Q) = 0. Если условие выполняется, то "отсекаем" рассматриваемую ветвь индекса; если нет, то переходим к проверке дочерних схем. Описанные действия применяем последовательно к полученным дочерним вершинам до тех пор, пока не будут рассмотрены все ветви индекса. После обхода дерева получим множество схем (являющихся листьями индекса), для которых Qs(S, Q) ≠ 0. Все документы, соответствующие оставшимся листовым схемам, исключим из пространства поиска, поскольку они не удовлетворяют запросу.

1.2. Индексирование

Описанный в предыдущем подразделе алгоритм поиска использует иерархию схем, однако не определяет способа ее построения. Для того чтобы сделать это, будем основываться на идее минимизации средней сложности вычислений, проводимых при выполнении алгоритма 1.1.

Таким образом, необходимо ввести функцию, характеризующую сложность вычислений запроса на схеме $Cost : \mathcal{S} \times \mathcal{Q} \to \mathbb{R}$.

1.2.1 Стоимость индекса

В процессе реального функционирования поисковая программа получает запрос извне. Однако в силу того, что невозможно описать все факторы, влияющие на то, какие запросы придется обрабатывать поисковой программе, будем полагать, что запрос программа получает случайно.

Рассмотрим действия алгоритма 1.1 на каком-либо запросе Q и индексе I. Вычислим стоимость усечения пространства поиска для выбранных индекса и запроса. Обозначим ее $Cost(Q, I)$.

Поскольку в рамках введенной модели множество запросов конечно, далее будем считать, что для $\mathcal{Q}$ задано вероятностное пространство запросов и оно определено следующим образом.

Тройку $(\Omega, \mathcal{F}, \mathcal{P})$ будем называть вероятностным пространством запросов, если:

- множество элементарных событий Ω есть множество запросов $\mathcal{Q}$;
- сигма-алгебра $\mathcal{F} = 2^{\mathcal{Q}}$;
- сигма-аддитивная мера задана через вероятности запросов естественным продолжением на всю сигма-алгебру.

Поскольку функция $Cost(Q, I)$ задана для всех элементарных событий, то она является случайной величиной. Таким образом, для заданного вероятностного пространства определена величина математического ожидания стоимости вычисления данного запроса. Именно этой величиной и воспользуемся для того, чтобы сравнивать индексы. Математическое ожидание стоимости усечения пространства поиска по данному индексу I назовем *стоимостью индекса I.*

Сумму по всем схемам индекса, кроме корневой, будем обозначать как $\Sigma_{S \in I}$. Величина $M(Cost(S, Q))$ характеризует среднее значение стоимости вычислений на схеме. В контексте вычисления оценок сложности будем обозначать ее $|S|$ и называть размером схемы. Вероятность того, что по схеме будет что-то найдено, назовем вероятностью схемы и будем обозначать $P\{\hat{S}\}$.

Тогда стоимость индекса

$$M(I) = \sum_{S \in I} P\{\hat{S}\}|S|.$$

Заметим, что стоимость индекса ограничена снизу. С учетом того, что конечны множества запросов, документов и схем, число различных индексов, которые можно построить для набора документов, также конечно. Таким образом, для каждой заданной базы данных существует оптимальный индекс I_0 такой, что для любого I верно $M(I) \geqslant M(I_0)$.

1.3. Построение оптимальных индексов

Предлагаемый в данном разделе алгоритм построения индекса имеет эвристический характер. Применим методику, аналогичную разбиению на

две равные части, используемую при построении оптимальных деревьев поиска [2]. Будем перебирать некоторые разбиения группы схем на две части с тем, чтобы выбрать наилучшее из них исходя из минимального значения стоимости индекса после разбиения.

Для разработки эффективного алгоритма построения индекса воспользуемся тем свойством, что для произвольной вершины оптимального индекса ветвь, состоящая из всех ее потомков, также является оптимальным индексом.

Таким образом, будем разбивать ветви индекса на две, начиная от корневой вершины. На каждом этапе будем перебирать разбиения ветви индекса и выбирать из них оптимальное. Переходя к дочерним вершинам, будем применять к ним аналогичные рассуждения.

Число всех возможных разбиений ветви, состоящей из N вершин, составляет 2^N. С тем чтобы ограничить количество вариантов меньшим числом, будем рассматривать центроиды для каждой схемы, как это изложено в работе [3]. Для каждой пары схем вычисляются расстояния между ними. Для каждой схемы S_i и каждого $l < N$ выбираются разбиения, состоящие из множества l ближайших к S_i схем, а также множества оставшихся схем. Такой подход позволяет сократить перебор до N^2 вариантов.

Для сокращения сложности вычисления значений $|S_a|$, $|S_b|$, $P\{S_a\}$ и $P\{S_b\}$ применяются оценки этих величин.

Алгоритм 1.2. ***Алгоритм построения индекса для поиска в наборе документов.***

Алгоритм представляет собой итерационный процесс. На каждом шаге обрабатывается вершина индекса из множества {S} ожидающих обработки вершин. Его суть в кратком изложении может быть представлена следующим образом.

- *Каждому документу из базы сопоставляем минимальную схему, его содержащую.*
- *Строим индекс, корнем которого является тривиальная схема, а листьями являются схемы, соответствующие документам базы.*
- *Добавляем в множество {S} корневую вершину индекса.*
- *Строим приближение M_S.*
- *ЦИКЛ, пока множество {S} не пусто.*

— Удаляем из множества ожидающих обработки вершин произвольную схему, обозначим ее S.

— Вычисляем расстояния между каждыми двумя схемами S_i', S_k' из множества S_1', ..., S_M', дочерних для S_1.

— ЦИКЛ по схемам S_j' из S_1', ..., S_M'.

** Упорядочиваем схемы S_1', ..., S_M' по возрастанию расстояния до S_j'.*

** Цикл по l от 1 до M − 1.*

** Строим множество из первых l ближайших к S_j' схем.*

** Вычисляем оценку для стоимости иерархии после разбиения $\{S_i', ..., S_k'\}$ на две части: множество l ближайших к S_j' схем и множество оставшихся дочерних для S_1 схем.*

— КОНЕЦ ЦИКЛА.

— Выбираем такое разбиение, оценка стоимости иерархии для которого минимальна.

— Разбиваем дерево индекса на части в соответствии с выбранным вариантом.

— Добавляем в S все вершины, которые стали для нее дочерними в {S}.

- *КОНЕЦ ЦИКЛА*

Отметим, что для реализации указанного алгоритма используются следующие функции над схемами:

- $|S| = \sum_Q Cost(Q, S)P\{Q\}$;
- $P\{S\} = \sum_{Q(S)\neq 0} P\{Q\}$;
- $S1 + S2$.

Необходимо учитывать, что сложность вычислений этих функций, в соответствии с их определениями, пропорциональна числу возможных запросов. С учетом изложенного при большом числе запросов вычисление этих функций не будет эффективным. Таким образом, представленная модель не позволяет эффективно вычислять подобные функции в общем виде. Однако на практике возможно реализовать вычисления более эффективно для заранее заданных моделей данных и вероятностных пространств запросов [1]. Как следствие, при реализации вычислений подобные функции необходимо рассматривать как внешние.

Опишем функции над $\mathcal{D}$, $\mathcal{S}$, $\mathcal{Q}$, на базе которых представляется возможным построить систему, позволяющую проводить индексацию документов и поиск в базе данных при использовании индексов.

Будем считать, что для множеств $\mathcal{D}$, $\mathcal{Q}$, $\mathcal{S}$ задана модель оптимизированного поиска, если заданы изоморфизм $\mathcal{S} \to 2^D$ и вероятностное пространство $(\Omega, \mathcal{F}, \mathcal{P})$, а также формально определены следующие функции:

- построение схемы по документу — $S(D)$;
- вычисление размера схемы — $|S|$;
- отношение на схемах — $S_1 \geqslant S_2$;
- объединение схем — $S_1 + S_2$;
- вычисление вероятности схемы — $P\{S\}$;
- вычисление запроса на документе — $Qd(D, Q)$;
- вычисление запроса на схеме — $Qs(S, Q)$;
- проверка соответствия документа схеме — $S \geqslant D$.

При этом для заданных функций выполняются следующие свойства:

- отношения $S_1 \geqslant S_2$, $S \geqslant D$ и операция $S_1 + S_2$ соответствуют заданному изоморфизму множества $\mathcal{S}$ и подмножества $2^{\mathcal{D}}$ посредством функции $\geqslant$;
- если верно $Qs(S, Q) = 1$, то существует $D \in S$: : $Qd(D, S) = 1$;
- если верно $Qs(S, Q) = 0$, то для любого $D \in S$: : $Qd(D, S) = 0$.

- функции $P\{S\}$ и $|S|$ заданы для вероятностного пространства Ω.

2. Структура системы

Исходя из базовых положений теоретической модели и заявленных функций опишем основные компоненты программной системы поиска и индексирования данных (далее сокращенно "система") и их взаимосвязи. Поставленная задача приводит к необходимости вынести описание внутренней структуры документов, схем и запросов в отдельные компоненты. Предлагаемая далее архитектура системы будет использоваться для разработки ее объектно-ориентированного кода на C++. Таким образом, компонентам системы будут соответствовать классы.

Документ, схема и запрос будут описаны в виде абстрактных классов — интерфейсов. Полностью описывая взаимодействие внешних классов с поисковой системой, такой подход позволяет отделить реализацию структуры документа, схемы и запроса от реализации алгоритмов поиска и рассматривать соответствующие задачи независимо.

Необходимо учитывать, что часть операций над схемами формально зависит от вероятностного пространства запросов. В связи с этим обстоятельством следует предусмотреть возможность изменять вероятностное пространство запросов, оставляя неизменными алгоритмы вычисления вероятности и размера схемы. На практике такая возможность предоставляется за счет того, что параметры, относящиеся к вероятностному пространству, хранятся в отдельном классе.

Поскольку вероятностное пространство не используется явным образом в алгоритмах поиска и индексации, соответствующий класс заворачивается в класс, который объединяет в себе информацию, единую для всех документов базы данных ("параметры базы данных").

Следует отметить, что формальная модель системы использует понятие базы данных как набора документов, однако в рамках поисковой системы база данных сама по себе не существует. С ней ассоциируются индекс, вероятностное пространство и другие параметры базы данных. Класс, состоящий из объектов перечисленных типов, будем называть экземпляром базы данных.

Будем считать, что поисковая система состоит из набора экземпляров баз данных. Каждый экземпляр состоит из индекса и базы данных.

Перечислим основные компоненты системы. Каждому ранее введенному понятию поставим в соответствие класс на языке C++:

- документ — abstract Document;
- схема — abstractScheme;
- запрос — abstractQuery;
- база данных — base;
- параметры базы данных — abstractParams;
- экземпляр базы данных — dataBaseInstance;
- индекс — index;
- поисковая система — searchServer.

Основные функции, которые должен выполнять searchServer при взаимодействии с внешней средой заключаются:

- в построении индекса по базе данных;
- в поиске информации в базе данных по запросу с использованием индекса.

Кроме основных функций в поисковой системе используются также функции вспомогательные. К их числу относятся те, которые основываются только на операциях ввода—вывода для документов, запросов и схем, например, запуск, остановка сервера; создание базы данных по ресурсу, содержащему документы; удаление базы данных, а также ряд других. Такие функции не требуют реализации каких-либо специальных алгоритмов, представляющих интерес в контексте настоящей публикации.

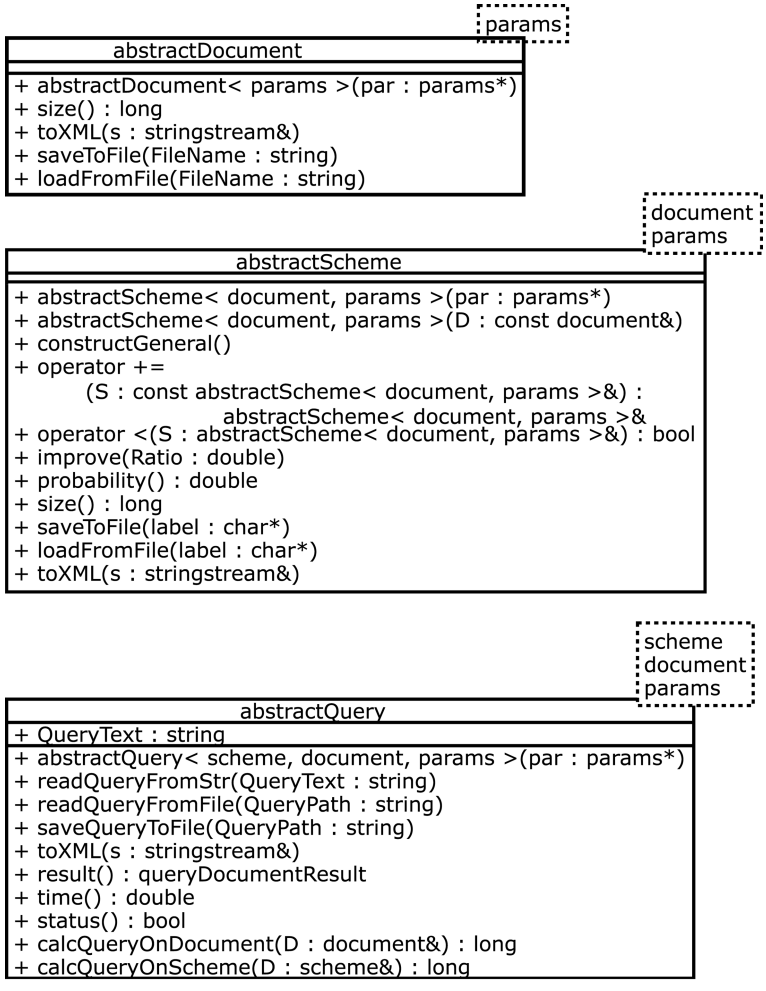


Рис. 1. UML-диаграммы интерфейсов документа, схемы и запроса

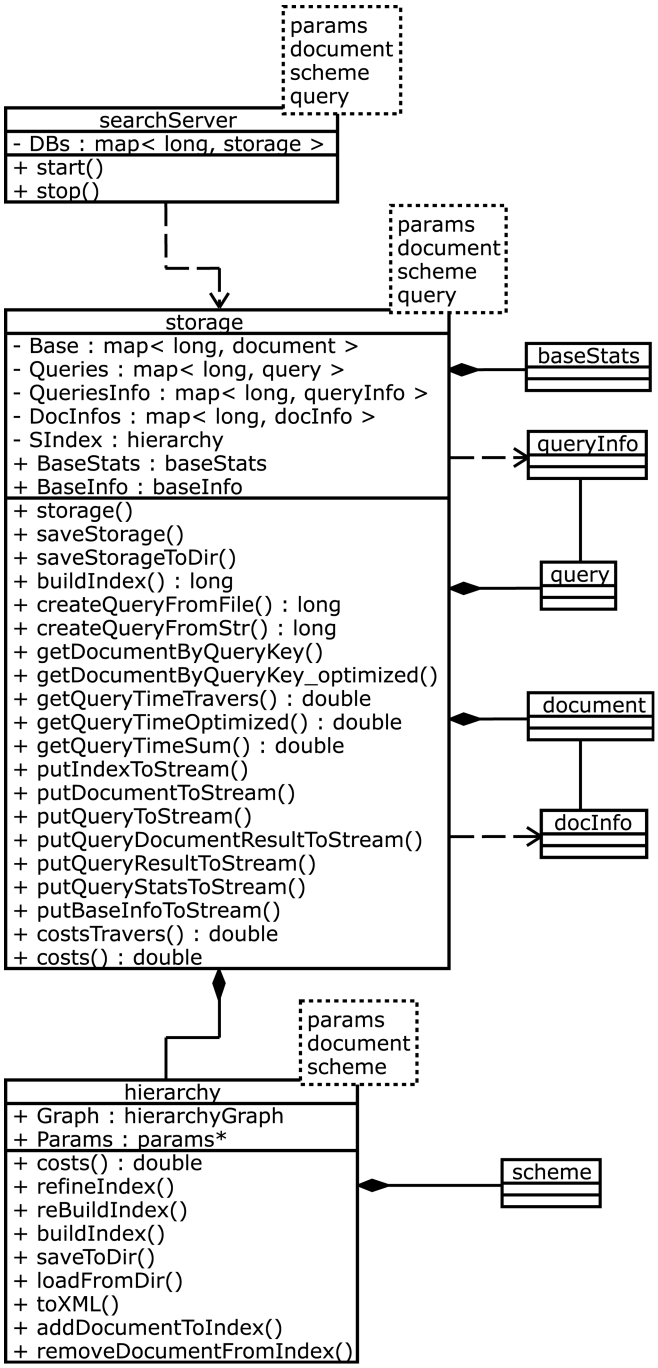


Рис. 2. UML-диаграмма сервера

Будем считать, что для классов abstractDocument, abstractScheme и abstractQuery задана формальная модель оптимизированного поиска, для них должен быть задан необходимый набор функций, описанный в п. 1.3. Если к этим функциям добавить вспомогательные, то получим классы abstractDocument, abstractScheme и abstractQuery, реализация которых позволит создать поисковую систему с заданным пользовательским интерфейсом.

На рис. 1 в виде UML-диаграмм представлены интерфейсы классов abstractDocument, abstractScheme и abstractQuery, описанные в рамках создания прототипа поисковой системы.

На рис. 2 представлены классы, с которыми связан searchServer: base, abstractParams, dataBaseInstance, index.

Предложенный в рамках настоящей работы подход программно реализован в виде прототипа системы на языке C++ с использованием шаблонов. Внешние интерфейсы документа, схемы и запросы реализованы в виде абстрактных классов.

При наличии разработанной системы процесс создания нового экземпляра системы для поиска в той или иной модели данных сводится к следующему. Необходимо создать три класса, реализующие абстрактные классы abstractDocument, abstractScheme, abstractQuery, описывающие, таким образом, модель данных. Затем следует объявить шаблонный класс сервера, зависящий от заданных классов.

С учетом изложенного задача создания сервера сводится к определению функций для документа, схемы и запроса. На основе описанного подхода реализован прототип системы поиска в базах данных на основе OEM-модели.

3. Оценки сложностей алгоритмов поиска и индексации

Опишем оценки сложностей алгоритмов поиска 1.1 и построения индекса 1.2.

Утверждение 3.1. *Предположим, что база данных состоит из N документов, а размер схемы оценивается сверху как $|S|$. Тогда сложность алгоритма построения индекса для поиска в наборе документов можно оценить сверху как*

$$O(NC_{|S|} + N^2(C_{S_1+S_2} + C_{P\{S\}}) + N^3\ln(N)).$$

Кроме описанного выше алгоритма построения индекса можно предложить ряд других (также эвристических), оценка сложности которых меньше на порядок или два (относительно N). Поскольку алгоритмы эвристические, сравнивать результат их работы следует в процессе испытаний на конкретных практических приложениях. Например, для модели данных, задающей поиск по уникальному идентификатору, можно предложить эффективный алгоритм построения индекса с оценкой сложности $N\ln(N)$. Однако для поиска в полуструктурированных данных по произвольным регулярным запросам подобный алгоритм будет неэффективен.

Возвращаясь к оценке сложности алгоритма поиска, отметим, что ей соответствует стоимость индекса, которая является математическим ожиданием стоимости вычислений при поиске по заданному индексу. В общем случае можно представить нижнюю и верхнюю оценку стоимости поиска.

Утверждение 3.2. *Предположим, что мощность множества запросов равна M, все запросы равновероятны, при этом все результаты их вычисления различаются между собой. Тогда не существует индекса (в определенном настоящей работой смысле),*

стоимость которого меньше ln(M). При этом существует алгоритм поиска, сложность которого можно оценить сверху как O(ln(M)).

Данное утверждение также некоторым образом показывает достижимость построения оптимального индекса для заданной базы данных. Однако, и это следует отметить, оно не применимо на практике, так как для такого построения необходимо вычислить все допустимые запросы и одновременно хранить все их результаты.

Выводы

Описанный в настоящей публикации подход позволяет в общем виде подойти к решению задачи построения оптимальных индексов для баз данных, представляющих собой наборы однотипных документов и, как следствие, применять оптимальные индексы для широкого класса задач, в частности, для случаев поиска в реляционных базах данных, полуструктурированных базах OEM-документов, а также поиска в наборах XML-документов.

Как факт, иллюстрирующий применимость предложенных решений на практике, следует отметить, что на основе теоретической модели был создан прототип поисковой систем в наборах OEM-документов, а также формализована модель поиска в наборах XML-документов. Прототип успешно прошел тестирование на представительном наборе документов.

Предложенные решения наиболее перспективны в задачах, где наиболее критичны время разработки, а также гибкость используемых подходов. В качестве возможных сфер применения можно отметить области интеграции разнородных информационных источников, поиск в полуструктурированных данных и Интернет.

Список литературы

1. **Горелов С. С., Васенин В. А.** Усечение пространства поиска в полуструктурированных базах данных при помощи иерархии схем документов // Программирование. 2005. Т. 6. С. 41—55.

2. **Walker W., Gotlieb C.** A top-down algorithm for constructing nearly optimal lexicographic trees // InGraph Theory and Computing. 1972. P. 303—323.

3. **Горелов С. С.** Оптимальные иерархии схем для поиска по конъюнктивным регулярным путевым запросам в полуструктурированных базах данных // Программирование. 2006. Т. 4. С. 38—56.

4. **Bray T., Paoli J., Sperberg-McQueen C. M. et al.** Extensible markup language (xml) 1.0 (fourth edition). 16 August 2006. http://www.w3.org/TR/2006/REC-xml-20060816/.

5. **Fernandez M., Malhotra A., J. Marsh et al.** Xquery 1.0 and xpath 2.0 data model. 2003. http://www.w3.org/TR/xpath-datamodel/.

6. **Kwong A., Gertz M.** Schema-based optimization of xpath expressions. 2002. citeseer.ist.psu.edu/kwong02schemabased.html.

7. **Новак Л. Г., Кузнецов С. Д.** Свойства схем данных xml // Тр. Института системного программирования. 2003. Т. 4.

УДК 004.652.4(045)

Полищук Ю. В., канд. тех. наук, инженер,
ООО "Волго-Уральский научно-исследовательский институт нефти и газа"
Черных Т. А., аспирант,
Оренбургский государственный университет

Моделирование подсистем хранения информации, ориентированных на хранение квазиструктурированных объектов

Рассмотрены наиболее распространенные модели хранения объектов в реляционных базах данных. Предложена модель хранения квазиструктурированных объектов в реляционной базе данных, основанная на применении технологии XML. Сформулированы преимущества использования разработанной модели хранения объектов.

Ключевые слова: *хранение информации, квазиструктурированная информация, автоматизированные информационные системы.*

Современный уровень развития информационных технологий позволяет накапливать и обрабатывать данные, объемы которых выражаются эксабайтами. Это дает возможность в полном объеме сохранять информацию о различных процессах, происходящих в длительные интервалы времени.

В связи с этим невозможно представить себе солидную организацию или предприятие, не использующие в своей работе автоматизированные информационные системы.

Классической методикой проектирования баз данных является создание отдельной таблицы для каждой описываемой сущности, затем в процессе нормализации — выделение отдельных таблиц для хранения атрибутов сущности (таблицы-справочники). Такой подход удобен при реализации баз данных с относительно небольшим числом описываемых объектов и при несложных и статичных связях между ними. Изменение структуры хранимых данных требует изменений в структуре таблиц, что приводит к полной перестройке всей системы [1].

Кроме того, на этапе проектирования возникают проблемы, обусловленные квазиструктурированностью данных и отсутствием явных связей между ними. Под квазиструктурированными данными обычно понимают информацию, в которой можно выделить некую структуру,

однако структура эта заранее целиком или частично неизвестна, либо может меняться с течением времени [2, 3].

Описанное выше показывает актуальность разработки модели структуры данных, не требующей переделок при появлении новых сущностей и позволяющей хранить произвольную информацию. Такая универсальная подсистема хранения менее эффективна, чем специализированная. Однако возможно создание решения, сочетающего приемлемую производительность и простоту с достаточной степенью универсальности.

Разрабатываемая подсистема хранения данных опирается на следующие основные принципы [1]:

- каждая сущность, информация о которой хранится, — это объект;
- каждый объект уникален и имеет уникальный идентификатор;
- объект имеет свойства (строковые, числовые, временные, перечислимые и т. д.), которые описывают атрибуты сущности;
- объекты могут быть связаны между собой произвольным образом;
- объект может быть хранилищем, в этом случае допускается хранение в нем других объектов.

Такая подсистема хранения данных не привязана к модели и позволяет реализовать практически любую логику.

Существуют различные модели хранения объектов. Большинство из них построено с использованием древовидной структуры. В качестве узлов дерева выступают объекты.

Остановимся подробнее на наиболее распространенных моделях хранения объектов.

Одна из наиболее простых моделей хранения объектов описывается в работе [2]. Эта модель базируется на ER-диаграмме, схема которой изображена на рис. 1.

Таблица описания классов объектов classes состоит из трех атрибутов и определяет иерархию классов, необходимую для моделирования данных. Атрибут classes.id является идентификатором класса. Название класса хранится в атрибуте classes.name. Атрибут classes.id_parent определяет отношение наследования в иерархии классов.

Объекты хранятся в таблице objects. Каждый хранимый объект имеет уникальный идентификатор objects.id и ссылку на класс объекта objects.id_class.

В таблице link_type определяются допустимые связи между классами объектов. Атрибут link_type.id хранит идентификатор разрешаемой связи. Название связи хранится в link_type.name. Ссылки на связываемые классы объектов хранятся в атрибутах link_type.id_class1 и link_type.id_class2.

Для хранения связей между объектами используется таблица obj_links. Атрибуты obj_links.id_ obj1 и obj_links.id_obj2 ссылаются на связываемые объекты. Тип связи определяется в атрибуте obj_links.id_ link_type.

Атрибуты классов объектов определяются в таблице attribs. Идентификатор атрибута класса — attribs.id. Для каждого атрибута указывается ссылка на идентификатор класса attribs.id_class. Название атрибута определяется в attribs.name.

Для хранения значений атрибутов используется таблица attrib_val. Идентификатор значения атрибута определяется в attrib_val.id. Ссылки на идентификаторы атрибута и объекта определяются в attrib_val.id_attr и attrib_val.id_obj соответственно. Значение атрибута в строковом виде хранится в attrib_val.value.

Рассмотренная модель хранения объектов является базовой, так как при такой организации объект может хранить лишь строковые атрибуты.

При необходимости модель можно расширить, добавив таблицы, позволяющие реализовать хранение атрибутов других типов.

Н. А. Банников предлагает следующую модель хранения объектов по технологии СТИКРИЗ [4], ER-диаграмма которой представлена на рис. 2.

Все атрибуты объекта хранятся в базе данных по группам согласно типу данных. Для атрибутов одного типа используется отдельная таблица. Связь атрибутов с объектами обеспечивается с помощью справочника, который состоит из двух таблиц. В первой таблице TYPES_LIST хранится описание типов данных: номер типа и его название. Во второй таблице PROPERTY_LIST хранятся свойства типа: номер типа, номер свойства, название свойства и низкоуровневый тип данных. В рассматриваемом примере низкоуровневый тип представляется перечислением: 1, 2, 3, 4, 5, так как в базе данных могут сохраняться пять типов данных: целые числа, указатель на строку базы данных, дробные числа, строки и текст. Хранение объектов реализовано в виде древовидной структуры, которая хранится в таблице DATA_LIST.

На рис. 3 представлена универсальная модель хранения объектов. Проанализировав ее принцип организации хранения данных, можно сделать вывод, что ее структура имеет общие корни с моделью хранения данных, представленной на рис. 2.

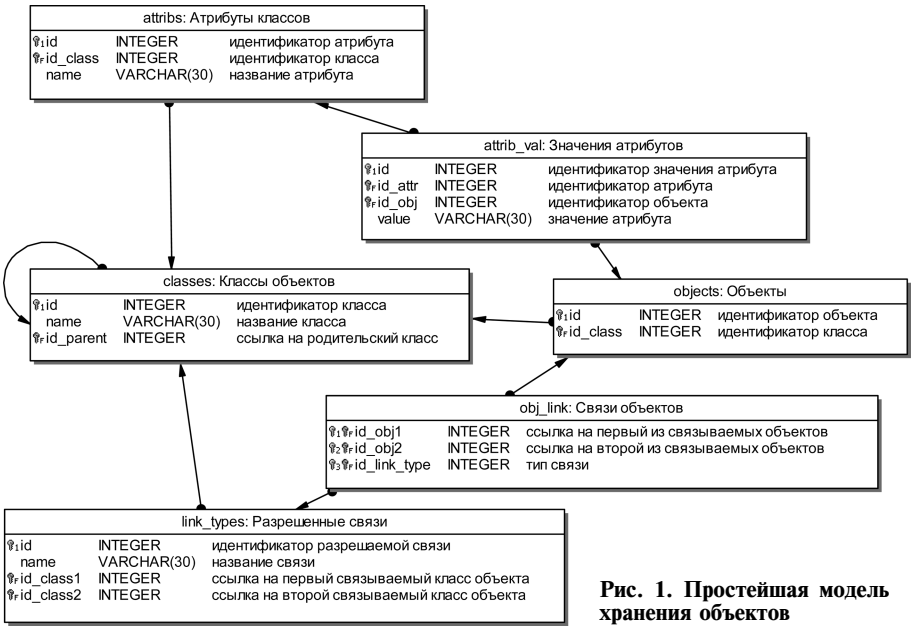


Рис. 1. Простейшая модель хранения объектов

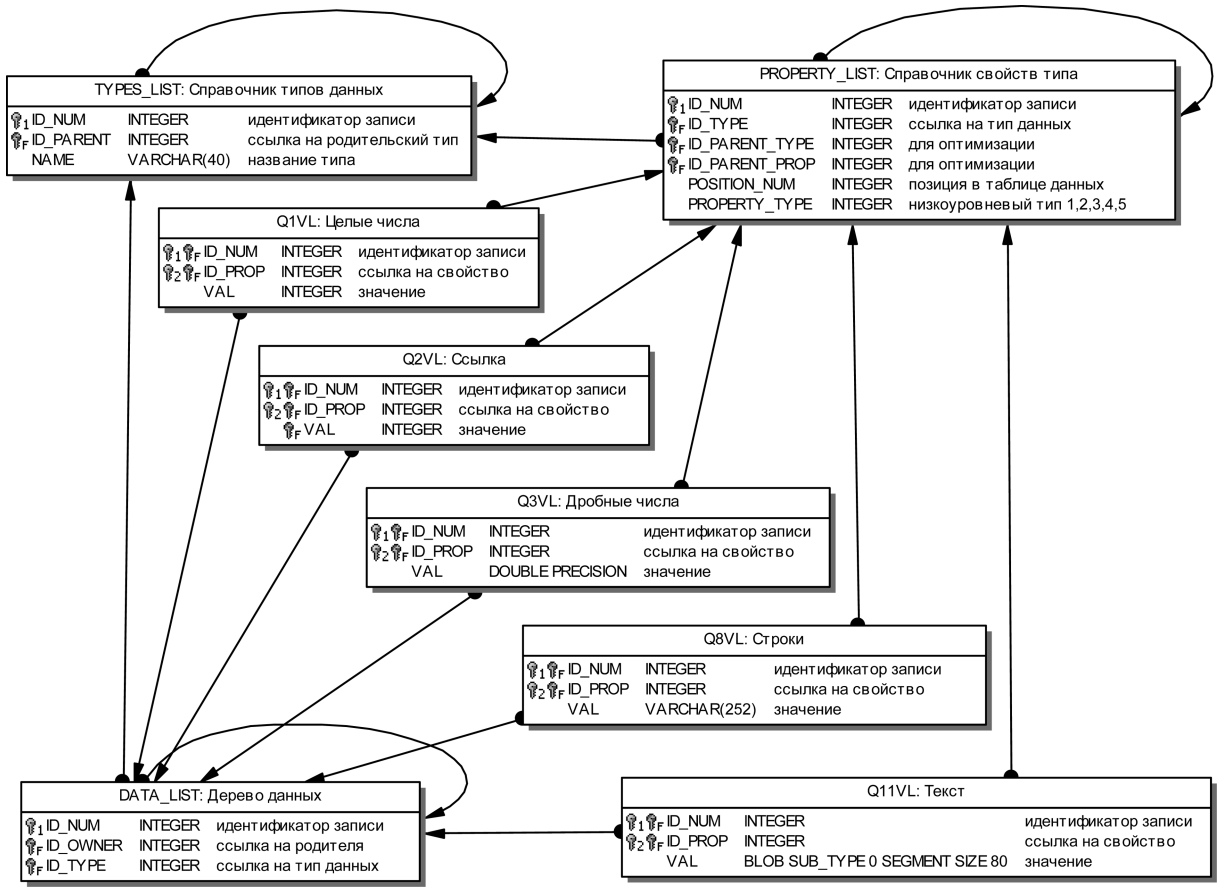


Рис. 2. Модель хранения объектов по технологии СТИКРИЗ

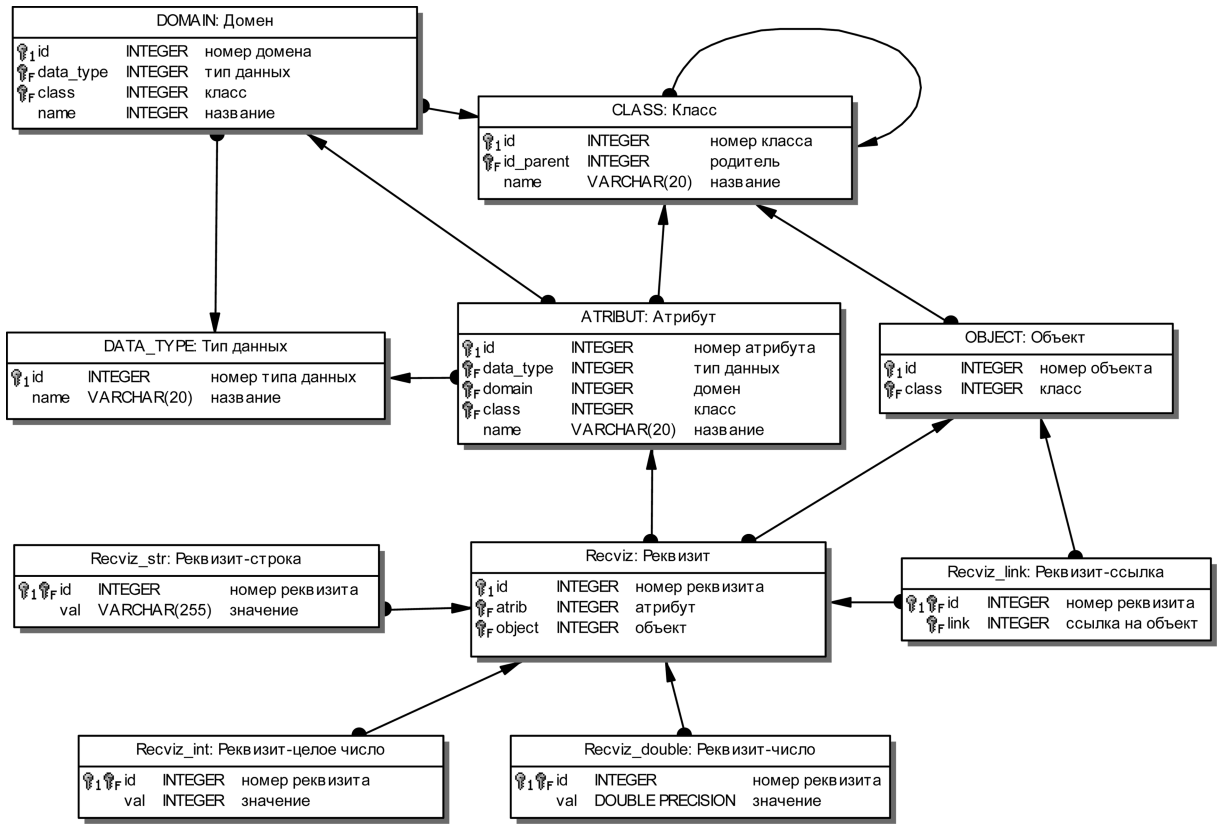


Рис. 3. Универсальная модель хранения данных

В этой модели хранения объектов, как и в предыдущей, каждый тип атрибута объекта хранится в отдельной таблице.

Рассмотрим модель данных, описанную в работе [1], ее Er-диаграмма изображена на рис. 4.

Представленная модель отличается от рассмотренных ранее возможностью хранения исторических и перечисляемых свойств объектов.

Таблица ObjType хранит информацию о типе объекта. Objects — таблица, в которой хранятся непосредственно сами объекты.

Описание строковых атрибутов хранится в таблице StrDesc, значения самих атрибутов хранятся в таблице Strings.

Для описания и хранения числовых атрибутов используются две таблицы: PropDesc и Properties.

Описание и хранение исторических атрибутов реализовано с помощью двух таблиц: Status и History.

Для описания и хранения перечисляемых атрибутов использованы три таблицы. Первая таблица EnumDesc задает, какие перечисления допустимы для выбранного типа. Вторая таблица EnumValues определяет возможные значения для перечислимого типа. Третья таблица Enums хранит непосредственно значения, связанные с объектом.

Связи между хранимыми объектами описываются с помощью трех таблиц. Типы связей определяются в таблице LinkType. Таблица AllowLinks совместно с триггером обеспечивает контроль возможных связей между объектами. Связи между объектами хранятся в таблице Links.

Развитие технологии XML позволило разработать новый подход к универсальной системе хранения данных. Второе поколение стандартов XML, в том числе и XML-схема, расширило границы применения технологии XML и позволило использовать их не только для обмена данными и инструкциями. XML-схема — первая модель данных, которая может быть использована для представления как неструктурированных "документов", так и структурированных "данных".

Schema Working Group международного консорциума W3C опубликовала спецификацию языка XML-схема в целях предоставления средств описания структуры, содержания и семантики XML-документов. Язык XML-схема можно рассматривать как улучшение DTD в том смысле, что XML-схема имеет строгую типизацию элементов и атрибутов, использует синтаксис XML для их описания. Его система типов довольно богата и насчитывает 47 скалярных типов данных, причем этот базовый набор типов данных может быть расширен более сложными типами с помощью таких методов, как наследование и расширение. Также поддерживаются последовательности и коллекции типов. Основанные на URN пространства имен используются для устранения неоднозначностей при формировании имен. XML-схемы документов могут иметь переменную структуру, обладающую необязательными атрибутами, необязательными и повторяющимися элементами, а также структурами выбора одного элемента из нескольких альтернативных.

Системы типов языка XML-схема достаточно для использования ее как модели хранения [5]:

- "структурированных" реляционных данных (т. е. когда структура каждого элемента является регулярной, коллекции элементов гомогенны, а периодически повторяющиеся элементы данных состоят из скалярных значений);

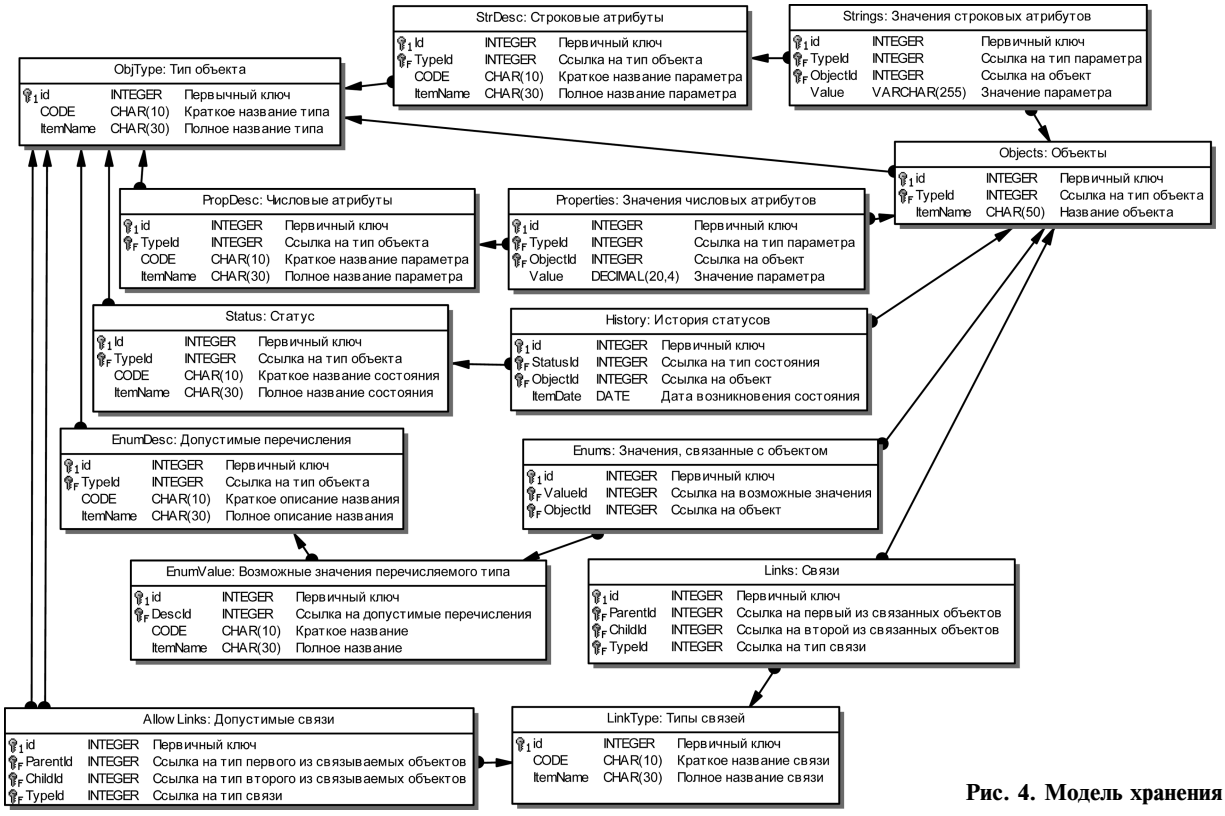


Рис. 4. Модель хранения объектов

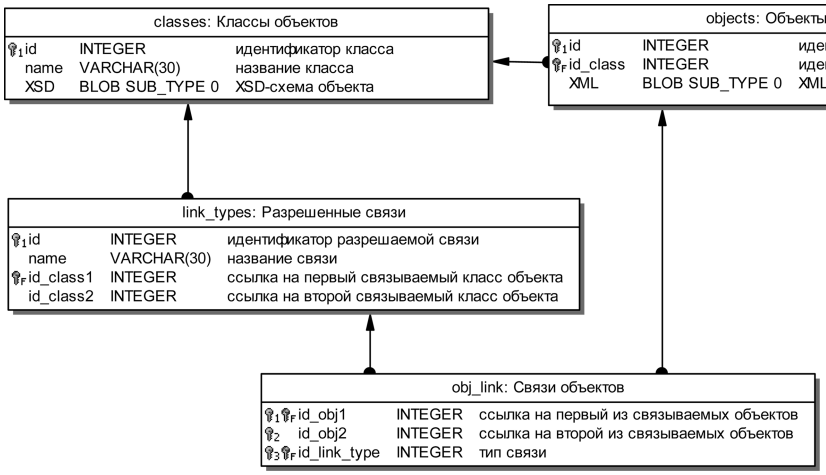


Рис. 5. Модель хранения данных с применением XML-схема

- "неструктурированных" документов (в которых структура является гибкой, а документ состоит из данных регулярной структуры и фрагментов нетипизированных примечаний или текста нерегулярной структуры);
- "полуструктурированных" (квазиструктурированных) документов (в которых структура хотя и есть, но меняется от одного экземпляра документа к другому).

Учитывая возможности модели данных XML, можно предложить следующую модель хранения квазиструктурированных объектов. Ее Er-диаграмма представлена на рис. 5.

В предложенной модели классы объектов хранятся в таблице Classes. Эта таблица состоит из трех атрибутов. Атрибут classes.id является идентификатором класса. Название класса хранится в атрибуте classes.name. Атрибут classes.XSD хранит XML-схему объектов класса.

Для хранения объектов используется таблица objects. Каждый сохраняемый объект имеет свой уникальный идентификатор objects.id и ссылку на класс объекта objects.id_class. Атрибут objects. XML хранит непосредственно сам объект.

Таблица link_types описывает разрешенные связи между классами хранимых объектов. Атрибут link_types.id определяет идентификатор разрешенной связи. Название разрешенной связи определяет атрибут link_type.name. Атрибуты link_types.id_classl и link_types.id_class2 хранят ссылки на связываемые классы объектов.

Связи между объектами определяются в таблице obj_link. Атрибуты этой таблицы obj_link.id_obj1 и obj_link.id_obj2 ссылаются на связываемые объекты. Тип связи определяется атрибутом obj_link.id_link_type.

Наполнение XML-объектов данными реализуется с помощью специального инспектора объектов, который, используя XML-схему объекта, контролирует корректность ввода данных.

Как уже отмечалось ранее, XML оптимально подходит для хранения документов. Следовательно, модель хранения данных, представленная на рис. 5, может быть положена в основу квазиструктурированного информационного хранилища, которое базируется на "полуструктурированных" объектах (документах).

Используя модель хранения данных с применением языка XML-схема, подсистему хранения можно построить в соответствии с двумя подходами к организации хранения информации [6]:

- по организационной структуре предприятия;
- по объектам работы организации.

При построении хранилищ данных организаций целесообразно применить оба подхода к организации хранения данных.

Таким образом, можно связать в единой модели хранения все данные, циркулирующие в организации.

При первом подходе информация распределяется по принадлежности к подразделениям, отделам, секторам и представляется в виде дерева в соответствии со структурой предприятия. Пример такой организации хранения информации представления на рис. 6.

При втором подходе информация по определенному объекту соотносится непосредственно с самим объектом. Все объекты, рассматриваемые организацией, должны быть по возможности структурированы и представлены в виде дерева. На рис. 7 представлен пример организации хранения информации по объектам работы месторождения.

Визуализация такого подхода к представлению информации позволит пользователям информационного хранилища наиболее эффективно ориентироваться в данных и будет способствовать их пониманию общей структуры работы организации. Также предложенный подход позволяет безболезненно масштабировать сис-

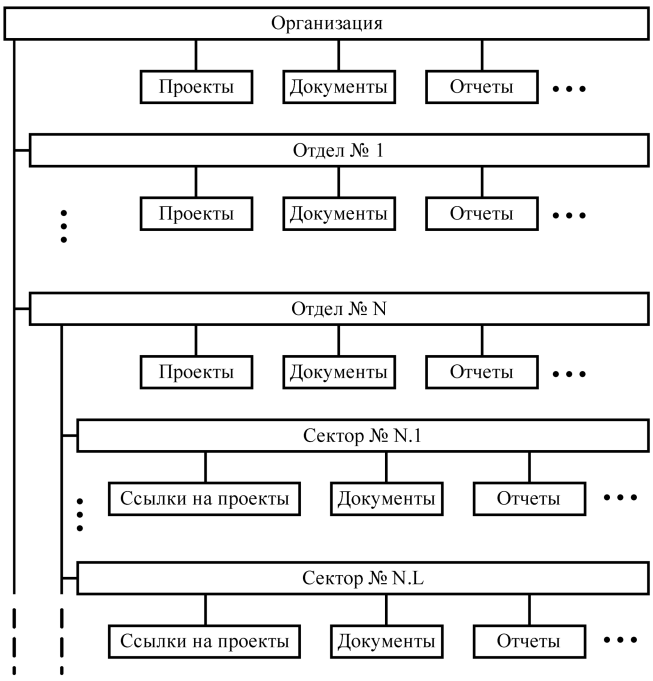


Рис. 6. Схема хранения информации в соответствии со структурой организации

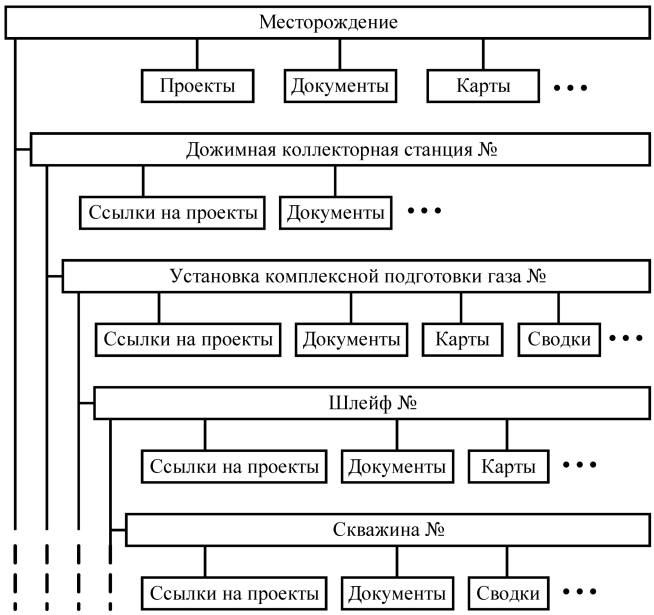


Рис. 7. Организация хранения информации по объектам работы

тему и обеспечивает возможность уточнения (дробления) или расширения общего информационного дерева без полной его переработки.

Функции поиска по хранилищу объектов должны быть реализованы с использованием возможностей интеллектуального анализа данных. Это позволит пользователям автоматизированной информационной системы формировать запросы к хранилищу объектов на естественном языке.

Кроме того, такая структура подсистемы хранения оптимально подходит для поиска и анализа данных, так как информация в ней представляется виде взаимосвязанных идентифицируемых объектов. Результатом поиска будет не просто информация по объекту, а указатели на объекты, размещенные в едином хранилище и связанные с другими объектами хранилища. Это позволяет пользователю ознакомиться с множеством связанных объектов, расположенных в структуре хранилища "вблизи" найденного объекта.

Рассмотренная объектно-ориентированная реляционная квазиструктурированная модель подсистемы хранения подходит для поэтапного развертывания информационного хранилища в организации и позволяет разрабатывать масштабируемые информационные хранилища.

Список литературы

1. **Тенцер А.** База данных — хранилище объектов // КомпьютерПресс. 2001. № 8. С. 144—145.
2. **Палей Д.** Моделирование квазиструктурированных данных // Открытые системы. 2002. № 09. С. 57—64.
3. **Григорьев Е.** Представление идентифицируемых сложных объектов в реляционной базе данных // Открытые системы. 2000. № 01...02. С. 55—62.
4. www.stikriz.narod.ru
5. **Sandeepan Banerjee.** Implementing XML Schema inside a 'Relational' Database. WWW2003, May 2003, Budapest, Hungary.
6. **Полищук Ю. В., Черных Т. А.** Особенности формирования информационно-справочной системы организации // Математическое и компьютерное моделирование в сложных системах: Сб. научн. тр. регионального научно-практического семинара. Оренбург: ИНПК ГОУ ОГУ, 2007. 146 с.

УДК 623.4.024

Н. И. Куренков, д-р техн. наук, проф., **С. Н. Ананьев,** канд. техн. наук, доц.,
3 ЦНИИ МО РФ

Критерий однородности матрицы и его использование в анализе многомерных данных

Предлагается новый подход к агрегированию многомерных данных на основе использования критерия однородности, определяемого как максимум показателя однородности одномерного массива. Сам показатель представляет собой отношение средней гармонической значений элементов массива к средней арифметической. Приводится обобщение этого определения на двумерный массив — матрицы как среднее гармоническое показателей однородности ее столбцов. Рассматриваются свойства показателя однородности, важнейшие из которых — инвариантность к преобразованиям подобия и инверсии. Эти свойства позволяют создавать эффективные алгоритмы агрегирования многомерных данных. Рассмотрены примеры использования предложенного критерия в теории принятия решений для построения интегрального показателя надежности банков и градации признаков в задаче оценки информативности частных показателей надежности банков. Исследования поддержаны РФФИ (грант № 05-08-65501).

Ключевые слова: *критерий однородности, показатель, матрица, обобщенная характеристика, преобразование инверсии, градация признаков.*

В научных исследованиях и при решении практических задач в медицине, биологии, технике, финансах, экономике и других областях исследователям часто приходится решать задачу классификации совокупности объектов предметной области по их многомерному описанию (например, для обнаружения похожих (близких) по структуре значений последовательностей при анализе временных рядов, распознавании слуховых образов, медицинской диагностике, диагностике технических систем и др.).

По своему содержанию указанная задача сводится к выявлению групп однородных по значениям массивов данных, представляющих собой описание (признаки) объектов. Эти описания могут задаваться соответствующими строками (векторами) признаков x_i ($i = \overline{1, n}$) матриц "объект—признак" X или матрицами "объект—объект" $R = (r_{ij})$ ($i, j = \overline{1, n}$).

Узловым моментом в решении такой задачи классификации является выбор критерия однородности анализируемых данных (объектов), согласно которому определяется принадлежность i-го объекта x_i соответствующей однородной группе (кластеру)[1].

Особенности мер однородности, используемых в существующих методах анализа многомерных данных

Традиционные методы классификации в зависимости от природы исходных данных в своем большинстве ориентированы на использование таких мер, как расстояние махаланобисского типа, являющееся обобщением эвклидова расстояния, коэффициент корреляции[2] или характеристики объектов, физически содержательные с точки зрения оценки расстояния (близости) между объектами и др. [1].

Замечательным свойством указанных мер является их инвариантность к аддитивным сдвигам, что позволяет исследовать структурные особенности признаков в окрестности их математических ожиданий при минимальном искажении геометрической структуры исходных данных. Именно этим во многом определяется эффективность и их широкое использование.

Однако на практике для реализации указанных желательных свойств приходится разрабатывать сравнительно сложные численные алгоритмы, необходимые для обеспечения корректного использования таких мер. Это означает выполнение целого ряда условий (требований), к которым относятся:

- невырожденность ковариационной (корреляционной) матрицы;
- обеспечение их малой изменчивости для различных классов исследуемых объектов (в дискриминантном анализе);
- желательность подчинения признаков (как случайных величин) нормальному закону;
- нормированность исходных данных.

Из перечисленных требований наибольшие сложности в выполнении, на наш взгляд, вызывает последнее, относящееся к данным, имеющим различные единицы измерения. Обусловлено это тем, что вид нормировки, выбор которой во многом субъективен, может приводить к неинтерпретируемости (противоречивости) результатов обработки [4]. Этот факт является следствием известной теоремы Подиновского, согласно которой изменение коэффициентов весомости признаков объектов (применительно к методу наименьших квадратов это эквивалентно изменению нормировки) может привести к изменению упорядочения этих объектов по средневзвешенному показателю [3].

Кроме того, ограничением большинства существующих методов автоматической классификации является их теоретико-вероятностные основания [1], согласно которым анализируемые исходные данные рассматриваются как выборка из некоторой генеральной совокупности. В соответствии с этой парадигмой порог меры однородности для объединения элементов выборки в кластеры определяется статистическими методами. Использование этой схемы для решения многих важных практических задач в условиях малого объема исходных данных (например, разработки прикладных систем, функционирующих в масштабе реального времени) оказывается затруднительным и даже неприемлемым вследствие невозможности получения представительных выборок.

В связи с этим с развитием средств вычислительной техники получили распространение новые методы анализа многомерных данных, не аппелирующие к их вероятностной природе [5]. Успех в разработке таких методов во многом зависит от реализации в них механизма выявления структурных особенностей признаков анализируемых выборок и использования этой информации для целей анализа.

Критерий однородности матрицы и его свойства

Предлагаемый критерий для одномерного массива представляет собой максимум показателя его однородности, имеющего вид отношения двух видов средних его элементов: средней гармонической и средней арифметической. Он определен для массивов с положительными значениями элементов и максимизируется в заданной условиями исследования параметрической области. Постулируется, что определяемые решением задачи максимизации параметры содержат в себе искомую информацию о структуре исходных данных, которую можно использовать в интересах исследования. Рассмотрим формальную схему вышесказанного утверждения.

Введем показатель однородности данных — одномерного массива положительных действительных чисел, представленных в виде вектор-столбца $x = (x_1, ..., x_n)^{\mathrm{T}}$, как квадратный корень из отношения средней гармонической к средней арифметической величине элементов массива для $\forall x \in R_+^n = \{r = (r_1, r_2, ..., r_n)^{\mathrm{T}} | r_i > 0 \; \forall i = \overline{1, n}\}$:

$$\theta(\boldsymbol{x}) = \frac{n}{\sqrt{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} \frac{1}{x_i}}}. \qquad (1)$$

Отметим, что (1) обладает следующими основными свойствами меры:

- симметрии ($\theta(x_1, x_2, ..., x_n) = \theta(x_{i_1}, x_{i_2}, ..., x_{i_n})$) для любой i-й перестановки компонент вектора x;
- монотонного убывания при добавлении к массиву $x \in R_+^n$ новых положительных элементов.

[1] Считается, что в задаче автоматической классификации понятие однородности является наиболее трудным и наименее формализованным [1]. Одной из причин такого положения является неопределенность в указании порога меры близости (однородности) сравниваемых объектов (расстояния между ними), при значениях меры ниже которого объекты можно считать однородными. В каждом конкретном случае этот порог определяется в зависимости от содержания задачи классификации.

[2] Например, коэффициента Пирсона для интервальных шкал, позволяющего представлять совокупность признаков гораздо меньшим их числом.

Мера удовлетворяет неравенству $0 \leqslant \theta(x) \leqslant 1$, причем $\theta(x) = 1$ тогда и только тогда, когда массив x состоит из одинаковых компонент.

Можно указать следующие особенности предлагаемого показателя.

Первая — инвариантность к мультипликативным константам или отношению подобия. Это позволяет не проводить нормировку данных, измеренных в различных единицах измерения и шкалах, и обрабатывать данные, представленные в шкале отношений, в которой структурные особенности признаков x_i более устойчивы, чем в других.

Вторая особенность заключается в использовании *преобразования инверсии* ($x \mapsto \frac{1}{x} \ \forall x \in R_+$), что позволяет разрабатывать простые и эффективные вычислительные алгоритмы, ориентированные на обработку существенно многомерных данных в масштабе реального (близком к реальному) времени.

Проверим работоспособность этого критерия на конкретных практически важных примерах.

Пример 1. В теории принятия решений одной из типовых задач является упорядочение объектов (альтернатив) из заданного конечного множества по степени важности. Широко известным методом решения этой задачи является метод анализа иерархий (МАИ), разработанным Т. Саати [6].

Суть этого метода заключается в следующем. Пусть задано множество критериев (факторов, акторов и т. п.) и множество объектов. Критерии и объекты структурированы в виде иерархической схемы (иерархии). Объекты и критерии необходимо попарно сравнить между собой по указанным критериям для оценки степени их влияния на глобальную цель исследования. Верхний уровень иерархии соответствует указанной глобальной цели, а самый нижний — сравниваемым объектам. Объекты (критерии) одного уровня попарно сравниваются между собой по критериям вышестоящего уровня с использованием специально выбранной девятибалльной шкалы. Процедура попарного сравнения объектов по связанному с ними критерию заключается в назначении экспертами оценки степени превосходства (приоритета) одного объекта над другим. Результат сравнения представляется в виде совокупности обратно симметричных матриц G, определяемых для каждого критерия каждого уровня иерархии [6]. Требуется выбрать объект, имеющий максимальное значение приоритета по отношению к цели.

Алгоритмическая основа метода базируется на вычислении векторов приоритетов для каждой обратно симметричной матрицы попарных сравнений в виде ее собственного вектора, отвечающего максимальному собственному значению. Несмотря на положительные свойства собственного вектора как вектора приоритета имеется неоднозначность, связанная с тем, что матрица парных сравнений G имеет два таких вектора. А именно, в качестве оценок вектора приоритетов могут использоваться не только компоненты правого собственного вектора $Gp = \lambda p$ (p — собственный вектор, отвечающий максимальному собственному значению λ), но также и нормированного к единице обратные значения $q^{-1} = \left(\frac{1}{q_1}, \frac{1}{q_2}, \ldots, \frac{1}{q_n}\right)^{\mathrm{T}}$ компонент левого собственного вектора $q = (q_1, q_2, \ldots, q_n)^{\mathrm{T}}$: $q^{\mathrm{T}}G = \lambda q^{\mathrm{T}}$ или $G^{\mathrm{T}}q = \lambda q$. Так как матрица G парных сравнений несимметрична, то указанные значения компонент векторов q^{-1} и p могут значимо отличаться друг от друга при коэффициенте отношения согласованности[3] (OC) больше некоторого порога γ: $OC > \gamma$ (обычно полагают $\gamma = 0{,}1 \ldots 0{,}2$). Какой из названных векторов лучше использовать для оценки вектора приоритета в этих условиях?

Вместе с тем, в соответствии с МАИ при больших значениях $OC > \gamma$ эксперту необходимо пройти процедуру согласования своего мнения, поскольку в его суждениях имеются противоречия, заключающиеся в нарушении свойства транзитивности его оценок. В рамках МАИ предлагается заново повторить всю процедуру парного сравнения. Очевидно, такой подход к согласованию мнения эксперта может занять много времени. Можно ли подсказать эксперту, с какой сравниваемой пары альтернатив целесообразно начать пересмотр своих суждений? Ответ оказывается утвердительным.

Для реализации этой "подсказки" необходимо отказаться от использования собственного вектора обратно симметричной матрицы парных сравнений G при определении вектора приоритетов и заменить ее на процедуру, использующую только инверсию элементов матрицы парных сравнений G. Предлагается реализовать такую процедуру определения искомого (нового) вектора приоритетов w в виде решения оптимизационной задачи:

$$w^{-\mathrm{T}}Gw \leftrightarrow \min_{w \in R_+^n} \tag{2}$$

для $R_+^n = \{p = (p_1, p_2, \ldots, p_n)^{\mathrm{T}} | p_i > 0 \forall i = \overline{1, n}\}$ (иногда, для краткости, вместо $p \in R_+^n$ будем писать $p > 0$).

Действительно, решение (2) обладает свойством обратной симметричности, поскольку имеет место равенство

$$(w^{-\mathrm{T}}Gw)^{\mathrm{T}} = w^{\mathrm{T}}G^{\mathrm{T}}w^{-1}, \tag{3}$$

которое означает, что можно решать либо задачу (2) либо задачу (4)

$$v^{-\mathrm{T}}G^{\mathrm{T}}v \to \min_{v \in R_+^n} \tag{4}$$

с поправкой на взятие обратной величины от полученного решения $w = v^{-1}$. Этот факт позволяет говорить об устойчивости решения (2) относительно операции инверсии. Заметим, что этим свойством не обладает вектор приоритетов, полученный на основе собственного вектора [6].

Для определения целевой функции в оптимизационной задаче (2) воспользуемся определением обратно симметричной матрицы $G = (g_{ij} > 0)$, для которой справедливо $g_{ij}g_{ji} = 1$, $\forall i, j = \overline{1, n}$. Следствием этого факта является соотношение

$$G^{-1} = G^{\mathrm{T}}, \tag{5}$$

[3] Отношение согласованности OC есть показатель степени непротиворечивости мнения эксперта и определяется как отношение индекса согласованности матрицы парных сравнений эксперта к случайному индексу (об индексе согласованности и случайном индексе см. ниже).

а сумма всех элементов матрицы G есть $e^{\mathrm{T}}Ge$ ($e = \underbrace{(1, ..., 1)}_{n}^{\mathrm{T}}$ — единичный вектор-столбец).

Следуя общей идее, будем искать вектор приоритетов $w \in R_+^n$ из условия, что он должен доставлять максимум мере однородности $\theta(\tilde{G}, w)$ обратно симметричной матрицы $\tilde{G} = G./(ww^{-\mathrm{T}})$, где операция (./) поэлементная,

$$\theta(\tilde{G}, w) \to \max_{w \in R_+^n}. \tag{6}$$

Решение задачи (2) эквивалентно решению задачи (6), поскольку для обратно симметричной матрицы G имеет место равенство

$$(\theta(\tilde{G}, w))^{-1} = (w^{-\mathrm{T}}Gw)(w^{\mathrm{T}}G^{\mathrm{T}}w^{-1}) = \\ = (w^{-\mathrm{T}}Gw)(w^{\mathrm{T}}G^{\mathrm{T}}w^{-1})^{\mathrm{T}} = (w^{-\mathrm{T}}Gw)^2 \tag{7}$$

(равенство $(w^{\mathrm{T}}G^{\mathrm{T}}w^{-1}) = (w^{\mathrm{T}}G^{\mathrm{T}}w^{-1})^{\mathrm{T}}$ есть следствие операции транспонирования скалярных величин).

На основании (6) решение задачи (2) (вектор w) можно интерпретировать как агрегат (обобщенную характеристику) строк матрицы G, поскольку при максимизации меры однородности посредством изменения вектора w отличительные особенности строк будут отражаться в значениях его компонент w_i. Именно поэтому и предлагается использовать решение задачи (2) в качестве вектора приоритетов для матрицы попарных сравнений альтернатив G.

Очевидно, что для согласованной матрицы G, т. е. матрицы, которую можно представить в виде $G = vv^{-\mathrm{T}}$ для некоторого вектора $v \in R_+^n$, имеет место

$$\min_{w>0} w^{-\mathrm{T}}Gw = \min_{w>0} w^{-\mathrm{T}}vv^{-\mathrm{T}}w = n^2 \text{ при } w = v. \tag{8}$$

Следовательно, решение w задачи (6) совпадает с вектором v для согласованной матрицы G, что соответствует смыслу вектора приоритетов.

Для сравнения матрицы G с согласованной матрицей вводится индекс согласованности IC, который определяется по формуле

$$IC = \frac{\lambda_{\max} - n}{n-1}, \tag{9}$$

где $\lambda_{\max}$ — максимальное собственное значение матрицы парных сравнений G, отвечающее собственному вектору p: $Gp = \lambda_{\max}p$. Отметим, что для согласованных матриц $G\lambda_{\max} = n$, отсюда следует, что значение их индекса согласованности равно нулю. Поэтому показатель IC служит мерой отклонения матрицы от согласованной матрицы.

В обозначениях решения задачи (6) IC можно представить в виде

$$IC = \frac{\lambda_{\max} - n}{n-1} = \frac{p^{-\mathrm{T}}Gp - n^2}{n(n-1)} \geqslant \frac{\min\limits_{p \in R_+^n} p^{-\mathrm{T}}Gp - n^2}{n(n-1)}. \tag{10}$$

Применительно к вектору приоритетов w, как решению задачи (2), нижнюю оценку IC примем за новый показатель согласованности, обозначив его как $\overline{IC}$:

$$\overline{IC} = \frac{(w^{-\mathrm{T}}Gw) - n^2}{n(n-1)}. \tag{11}$$

Новый индекс согласованности $\overline{IC}$, очевидно, является функцией от элементов матрицы G. Поэтому, если все матричные элементы — случайные величины, распределенные по равномерному закону со значениями шкалы сравнения, используемой в МАИ (от 1 до 9), то значение $\overline{IC}$ будет также случайной величиной. Индекс согласованности обратно симметричной матрицы $\overline{IC}$, сгенерированный случайным образом, назовем в соответствии с [6] случайным индексом ($\overline{RI}$). В обозначениях, сделанных выше, среднее значение $\overline{RI}$ можно определить в первой строке таблицы, полученной по выборке из 10000 наблюдений, сгенерированных случайным образом в 9-бальной шкале и усредненной по всему объему наблюдений. Для сравнения во второй строке этой таблицы представлены значения RI, используемого в МАИ (стр. 34 [6]). Как и следовало ожидать, значения $\overline{RI}$ меньше соответствующего значения RI, что следует из неравенства (10). Поскольку значения $\overline{IC}$ также меньше соответствующего значения IC в соответствующих пропорциях к случайным индексам, то для определения граничных значений для показателя $\overline{OC}$, аналога OC, можно оставить пороговое значение, равное 0.1.

По значениям отношения согласованности

$$\overline{OC} = \frac{\overline{IC}}{\overline{RI}} \tag{12}$$

можно выявить все пары альтернатив, экспертные оценки которых не согласуются с оценками эксперта остальных пар альтернатив.

Теперь рассмотрим значение целевой функции (2). Имеет место равенство

$$w^{-\mathrm{T}}Gw = w^{-\mathrm{T}}\frac{G^{\mathrm{T}} + G}{2}w = \\ = \frac{1}{2}\left(\sum_{1 \le i < j \le n} g_{ij}\frac{w_j}{w_i} + g_{ji}\frac{w_i}{w_j}\right) + n. \tag{13}$$

Следовательно, каждая пара альтернатив (i, j) определяет аддитивный вклад

$$v_{ij} = g_{ij}\frac{w_j}{w_i} + g_{ji}\frac{w_i}{w_j}. \tag{14}$$

По значениям этого вклада всегда можно определить степень близости альтернатив i и j, поскольку с точностью до константы величина v_{ij} есть квазимера. Ее минимальное значение достигается на равных оценках альтернатив. Симметричность очевидна, а что касается неравенства треугольника, то для конечного множества всегда можно добавить аддитивную константу (с сохранением первых двух свойств) так, чтобы это неравенство выполнялось.

Этот результат позволяет выделить те пары альтернатив (объектов), сравнительные оценки которых вносят

Средние значения случайных индексов

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\overline{RI}$	0,00	0,00	0,5	0,78	0,97	1,09	1,18	1,25	1,30	1,34	1,37	1,4	1,42	1,44	1,45
RI	0,00	0,00	0,58	0,9	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,57	1,59

наибольший вклад в значение $\overline{IC}$, а следовательно, и в отношение согласованности (12). Именно с них предлагается осуществлять пересмотр индивидуальных оценок экспертов в случае больших значений $\overline{OC}$ (12). Очевидно, этот способ позволит эксперту значительно сократить время экспертизы, что подтверждается многими примерами его практического использования [7].

Прежде чем перейти к рассмотрению примера 2 использования меры однородности матриц, необходимо привести (1) к соответствующему виду, а именно, распространим меру (1) на матрицы, состоящие из m вектор-столбцов $X = [x_1, x_2, ..., x_m]$. Если $\theta(x_j) = \theta_j$ — мера однородности j-го столбца x_j, то их средняя гармоническая величина $\frac{m}{\sum_{j=1}^{m} 1/\theta_j}$ после преобразования будет

$$\theta(X) = \left(\frac{1}{nm} \sum_{j=1}^{m} \sqrt{\sum_{i=1}^{n} x_{ij} \sum_{i=1}^{n} \frac{1}{x_{ij}}} \right)^{-1}. \qquad (15)$$

Использование средней гармонической величины для свертки мер однородности столбцов позволяет повысить чувствительность $\theta(X)$ к изменению малых значений элементов матрицы X. Предложенная мера однородности матрицы (15) позволяет перейти к решению задачи агрегирования элементов ее строк. Общая идея здесь состоит в следующем.

Пусть p_i — нормировка для i-й строки матрицы X, что означает следующее: все элементы этой строки делятся на значение $1/p_i$. Совокупность таких чисел обозначим как вектор-столбец $p = (p_1, p_2, ..., p_n)^{\mathrm{T}} \in R_+^n$. Результат нормировки матрицы X посредством вектор-столбца p применительно к мере (15) можно записать в матричном виде:

$$\theta(X, p)^{-1} = \frac{1}{nm} \sum_{j=1}^{m} \sqrt{p^{-\mathrm{T}} x_j x_j^{-\mathrm{T}} p}, \qquad (16)$$

где x_j — j-й столбец матрицы $X = [x_1, x_2, ..., x_m]$; $p^{-\mathrm{T}}$ и $x_j^{-\mathrm{T}}$ — массивы обратных значений элементов массивов p и x_j.

Квадратный корень в (16) усложняет аналитические выкладки. Избавиться от него можно, применив прием квазилинеаризации [8]. Очевидно, имеет место равенство: для $\forall a > 0,\ b > 0,\ p > 0$

$$\sqrt{ab} = 0{,}5 \min_q \left(\frac{a}{q} + bq \right). \qquad (17)$$

Подставляя (17) в (16) и проводя необходимые упрощения, получим

$$2nm\theta(X, p)^{-1} = \min_{q>0} (p^{-\mathrm{T}} X q^{-1} + p^{\mathrm{T}} X^{-1} q). \qquad (18)$$

Процедура агрегирования заключается в том, что путем нормирования матрицы X с помощью вектор-столбца p осуществляется максимизация (15) или, что одно и то же, минимизация правой части (18) по p:

$$p^* = \arg\min_{p>0} (\min_{q>0} p^{-\mathrm{T}} X q^{-1} + p^{\mathrm{T}} X^{-1} q) =$$

$$= \arg\min_{p>0} \sum_{j=1}^{m} \sqrt{p^{-\mathrm{T}} x_j x_j^{-\mathrm{T}} p}. \qquad (19)$$

Предлагается результат решения задачи (19) использовать в качестве агрегированного признака (обобщенной характеристики, сводного показателя) для строк матрицы X.

Аналогично (19) можно показать, что обобщенная характеристика столбцов есть решение задачи

$$q^* = \arg\min_{q>0} (\min_{p>0} p^{-\mathrm{T}} X q^{-1} + p^{\mathrm{T}} X^{-1} q) =$$

$$= \arg\min_{q>0} \sum_{j=1}^{m} \sqrt{p^{-\mathrm{T}} x_i x_i^{-\mathrm{T}} q}. \qquad (20)$$

Таким образом, решение задачи (p^*, q^*) дает способ определения обобщенных характеристик строк и столбцов для любой матрицы X, состоящей из положительных элементов:

$$(p^*, q^*) = \arg \min_{p>0,\, q>0} \theta(X, p, q) =$$

$$= \arg \min_{p>0,\, q>0} (p^{-\mathrm{T}} X q^{-1} + p^{\mathrm{T}} X^{-1} q). \qquad (21)$$

Легко видеть, что решение задачи (21) не зависит от способа нормирования столбцов и строк матрицы X. Этот факт можно использовать при агрегировании размерных величин.

В общем случае вес каждого признака может быть различным. Тогда, если h — вектор коэффициентов веса признаков, а $D(h)$ — диагональная матрица, образованная этим вектором, то рассуждая аналогично (19), получим

$$p^* = \arg\min_{p>0} (\min_{q>0} p^{-\mathrm{T}} X D(h) q^{-1} + p^{\mathrm{T}} X^{-1} D(h) q) =$$

$$= \arg\min_{p>0} \sum_{j=1}^{m} h_j \sqrt{p^{-\mathrm{T}} x_j x_j^{-\mathrm{T}} p}. \qquad (22)$$

Пример 2. Рассмотрим задачу построения сводного показателя надежности (рейтинга) банков на основе его частных показателей. Для этого желательно частные показатели надежности представить таким образом, чтобы их большие значения соответствовали лучшим банкам (соответственно меньшие — худшим). Для этих целей естественно использовать преобразование инверсии, поскольку мера однородности $\theta(x, p, q)$ инвариантна к такому виду преобразований. Отметим, что понятия лучший (худший) банк определяется экспертами. Здесь же речь идет о количественной мере этих качественных оценок.

В качестве частных показателей надежности банков примем показатели: степенной (KI) и показательный

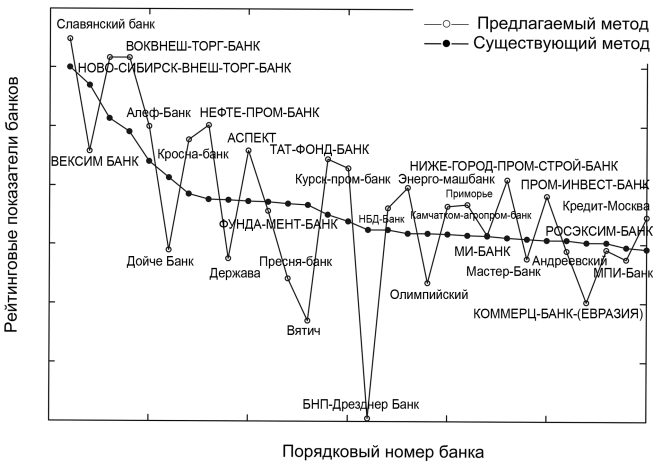


Рис. 1. Рейтинговые показатели банков, полученные существующим и предлагаемым методами

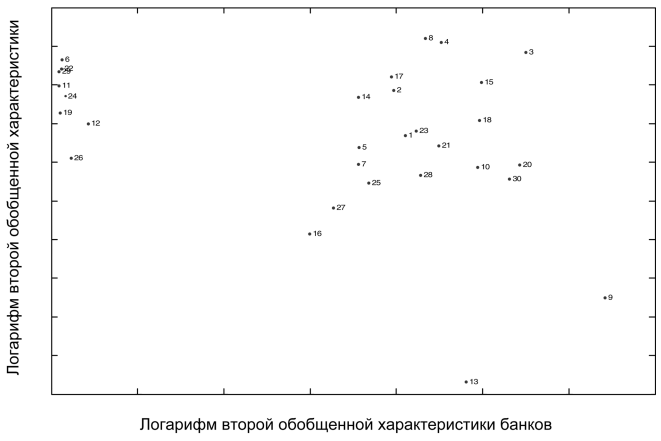


Рис. 2. Диаграмма рассеивания банков в плоскости двух обобщенных характеристик

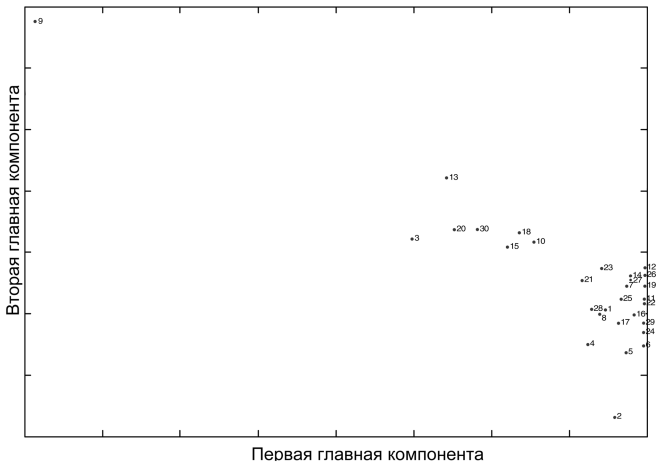


Рис. 3. Диаграмма рассеивания банков, полученная методом главных компонент

(*K2*) коэффициенты, коэффициент мгновенной ликвидности (*K3*), кросс-коэффициент (*K4*), генеральный коэффициент ликвидности (*K5*), генеральный коэффициент надежности (*K6*), коэффициент защищенности капитала (*K7*), коэффициент фондовой капитализации прибыли (*K8*), определение и методика расчета которых изложены в [9]. Там же показатель надежности банка Q предложено вычислять в виде линейной свертки

$$Q = \sum_{i=1}^{8} h_i K_i, \qquad (23)$$

где h_i — элементы вектора h = (0,15; 0,1; 0,175; 0,042; 0,275; 0,225; 0,025; 0,008) — результаты экспертных оценок. На рис. 1 представлены результаты расчетов рейтинга банков по формулам (22), (23), где видно, что диапазон изменения показателя (22) заметно больше, что говорит о его чувствительности к изменению малых значений частных показателей надежности банков.

Две обобщенные характеристики p^* и q^* позволяют представить совокупность банков в виде диаграммы рассеивания на плоскости (рис. 2). Для сравнения построим диаграмму рассеивания в плоскости двух первых главных компонент (рис. 3). Сравнение диаграмм показывает, что первая из них более точно определяет структурные особенности банков, поскольку выделяет не только общие кластеры, присущие обеим диаграммам, но и выделяет дополнительный кластер (№ № 26, 12, 19, 24, 11, 19, 29, 6), что объясняется большей чувствительностью обобщенных характеристик к малым значениям по сравнению с компонентами соответствующих собственных векторов.

Пример 3. Предложенную меру однородности можно с успехом использовать для градации массивов с положительными значениями. Данная задача относится к сложным, поскольку равномерное разбиение в общем случае не является оптимальным (в смысле определенного критерия). В [10] авторами предложено использование информационного критерия для решения этой задачи. Под градацией исходного одномерного массива $x = (x_i,\ i = \overline{1, n})$ будем понимать процедуру разбиения всех его значений на заданное число k непересекающиеся подмножеств элементов массива с близкими значениями. Для краткости все такие подмножества будем также называть градациями. Если $y = (y_i)$ — новый массив, составленный из средних значений g_i ($i = \overline{1, k}$) соответствующих градаций элементов массива x, то формальная постановка задачи оптимальной градации с использованием меры однородности (1) сводится к следующему:

$$\theta(x./y,\ g_1,\ g_2,\ ...,\ g_k) = \frac{n}{\sqrt{\sum_{i=1}^{n} \frac{x_i}{y_i} \sum_{i=1}^{n} \frac{y_i}{x_i}}} \to \max_{g_1,\ q_2,\ ...,\ g_k}. \qquad (24)$$

С использованием процедуры градации признаков можно проводить обработку многомерных разнотипных данных [11], в частности, оценивать информативность признаков различной природы. Для иллюстрации этой идеи оценим важность (вес) частных показателей надежности банков для примера 2. Эту важность определим как информативность частных показателей, необходимую для разбиения рассматриваемой совокупности банков в равных долях на группы: лучшие, средние, худшие, которые уже были определены по оценкам рейтинга банков, полученным существующим методом. Оценка информативности частных показателей банков включает процедуру их градации с использованием (24) и вычисления взаимной информации полученного градуированного признака с признаком требуемого разбиения $[\underbrace{1,...1}_{10},\ \underbrace{2,...2}_{10},\ \underbrace{3,...3}_{10}]^{\mathrm{T}}$ на основе методов алгебраической информации [12]. В результате оценки информативности I_i каждого i-го частного показателя надежности банков ($i = \overline{1, 8}$) был получен вектор оценок важности (информативности) I = [0,21; 0,16; 0,09; 0,17; 0,15; 0,1; 0,13] частных показателей K_i, которые целесообразно учитывать при определении рейтинга банков.

Очевидно, что предложенный метод оценки информативности частных показателей надежности банков применим для любого априорного разбиения их совокупности, поэтому всегда можно определить показатели, наиболее весомые с точки зрения целей исследования.

Список литературы

1. **Айвазян С. А., Бухштабер Б. М., Енюков И. С., Мешалкин Л. Д.** Прикладная статистика. Классификация и снижение размерности. М.: Финансы и статистика, 1989. 608 с.
2. **Леонов Ю. П.** Теория статистических решений и психофизика. М.: Наука, 1977. 233 с.
3. **Орлов А. И.** Теория принятия решений: Учебн. пос. М.: Изд-во "Март", 2004. 656 с.
4. **Куренков Н. И., Лебедев Б. Д.** Энтропийный анализ многомерных данных // Современные проблемы механики гетерогенных сред. Сб. трудов. М.: Изд. РАН. 2000.
5. **Малиновский Л. Г.** Анализ статистических связей: модельно-конструктивный подход. М.: Наука, 2002. 688 с.
6. **Саати Т.** Принятие решений: Метод анализа иерархий. М.: Радио и связь, 1993. 315 с.
7. **Куренков Н. И.** Развитие метода анализа иерархий в задачах обработки экспертной информации: Научно-метод. матер. исследований, труды семинаров, научно-технических конф. 3 ЦНИИ МО РФ. 2005. Вып. 7.
8. **Беккенбах Э., Беллман Р.** Неравенства. М.: Мир, 1965. 276 с.
9. **Черкасов Д.** Как мы считали надежность банков // Компания. 1999. № 7. Приложение.
10. **Ананьев С. Н., Куренков Н. И.** Информационный критерий и его использование в анализе многомерных данных // Информационные технологии. 2007. № 9.
11. **Лбов Г. С.** Методы обработки разнотипных экспериментальных данных. Новосибирск: Наука, 1981. 158 с.
12. **Гоппа В. Д.** Введение в алгебраическую теорию информации. М.: Наука. Физматлит, 1995. 112 с.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СОЦИАЛЬНЫХ И ЭКОЛОГИЧЕСКИХ СИСТЕМАХ

УДК 629.075

А. Ю. Переварюха, аспирант,
Санкт-Петербургский институт информатики и автоматизации РАН

Нелинейные модели и особенности оптимизации в задаче системного анализа динамики популяций

Предлагается непрерывно-дискретная математическая модель для оценки и прогнозирования эффективности воспроизводства промысловых популяций. Модель основывается на наличии пороговых эффектов в биологии развития, изучаемых в рамках теории этапности развития организмов. Топология фазового пространства динамической системы на основе разработанной модели качественно отличается от широко известных моделей Рикера и Бивертона—Холта. Делается вывод об опасностях, связанных с переводом популяции в состояние, наиболее оптимальное для промысла. Статья рассчитана на специалистов в области применения информационных технологий и имитационных моделей в экологии.

Ключевые слова: *моделирование динамики популяций; нелинейные динамические системы; оптимизация эксплуатации биоресурсов.*

Введение

Классические представления, господствовавшие ранее в естествознании, со второй половины XX века постепенно пересматриваются после того, как были открыты и исследованы системы, которые называют нелинейными. Еще в 1892 г. один из создателей качественной теории дифференциальных уравнений и теории бифуркаций Анри Пуанкаре показал на примере проблемы "трех тел" небесной механики возможность возникновения нестабильных орбит, которые существенно зависят от начальных условий и не могут быть точно вычислены. Илья Пригожин [1] приводит хороший образный пример: со времени первых работ по механике движение маятника изучалось с особой тщательностью. Если же расположить маятник так, чтобы груз оказался в точке, противоположной самому нижнему положению, то рано или поздно он упадет. В принципе невозможно предсказать, упадет он вправо или влево. Направление падения существенным образом зависит от незначительных флуктуаций. Верхнее неустойчивое положение маятника практически никогда не находилось в фокусе внимания исследователей.

В рамках нелинейной динамики изучается не только переход к детерминированному хаосу, выводящему фазовую траекторию за горизонт предсказуемости, но и способность сложных саморегулируемых систем (к которым относятся биологические сообщества) возвращаться в устойчивое состояние равновесия. Проблема эффективного использования природных ресурсов связана с проблемой стабильности биологических сообществ.

В настоящей статье речь пойдет об имитационных моделях воспроизводства популяций рыб, исследуемых в виде нелинейных динамических систем с гибридным представлением времени и разработанных с расчетом на применение современных вычислительных средств. Нелинейная динамика стала развиваться именно после появления возможности проведения численных расчетов на ЭВМ: Эдвард Лоренц исследовал таким способом не имеющую аналитического решения систему уравнений, что привело его к обнаружению знаменитого аттрактора.

Запас — пополнение: теория и модели

Концепции о линейной зависимости между запасом и пополнением Ф. Баранова и К. Бэра [2] не нашли

подтверждения в последующих исследованиях. Уравнение для численности пополнения (recruits) R предложено известным канадским исследователем У. Е. Рикером в 1953 г. [3]:

$$R = aS\exp(-bS), \qquad (1)$$

где S — (stock) нерестовый запас; b — коэффициент, отражающий величину, обратную количеству выметанной икры, при котором число выжившей молоди максимально (имеет смысл только $b \ll 1$); a — параметр.

До Рикера вопрос о возможности уменьшения пополнения при возрастании нерестового стада никем не рассматривался.

График зависимости числа рекрутов (по оси ординат) от численности производителей (ось абсцисс) называется кривой пополнения и представляет собой для (1) куполообразную кривую с единственным нетривиальным пересечением с биссектрисой координатного угла $R = S$. Очевидно, что при применении модели (1) (рис. 1) возникает ряд сложностей, например, при увеличении количества отложенной икры выживаемость молоди будет стремиться к нулю, что противоречит наблюдениям над аквариумными популяциями. Модель Рикера при имитационном моделировании популяций часто использовалась только до некоторого критического количества S, при превышении которого количество молоди определяется константой.

При крайне низких численностях производителей уравнение Рикера предсказывает увеличение эффективности воспроизводства популяции. При стремящемся к нулю количестве отложенной икры выживаемость пополнения стремится к предельному значению выживаемости:

$$\frac{dR}{dS} = a\exp(-bS)(1 - bS);$$

$$\lim_{S \to 0} a\exp(-bS)(1 - bS) = a.$$

Подобное не соответствует фундаментальным представлениям экологии о существовании нижней критической численности у популяций животных — принципу Олли, согласно которому существует оптимальный для воспроизводства диапазон численности репродуктивной части популяции, при уменьшении численности эффективность воспроизводства должна снижаться, так как уменьшается вероятность встречи особей разного пола в местах, пригодных для размножения.

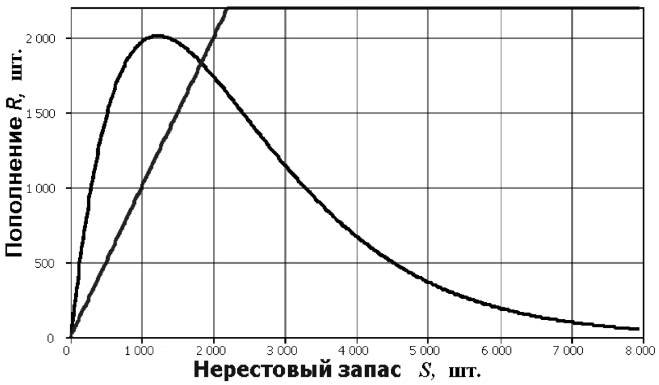


Рис. 1. Кривая запас—пополнение модели Рикера

У модели (1) есть и более интересные свойства, о которых не имел представления ее разработчик и многие из тех, кто ее использовал. Представим процесс изменения состояния популяции, определяемый зависимостью запас—пополнение в виде динамической системы: математического объекта, для которого можно указать набор величин — динамических переменных, характеризующих состояние системы. Значения таких переменных в последующий момент времени рассчитываются на основании текущих значений по определенному правилу (называемому оператором эволюции). Динамическая система — это тройка (M, T, ψ), состоящая из фазового пространства M, времени T, оператора эволюции ψ, причем для всех $x \in M$ и $t, s \in T$ выполняется условие

$$\psi(\psi(x, t), s) = \psi(x, s + t).$$

Множество $\{\psi^{(t)}(x)\}_{t \in T}$ называют фазовой траекторией точки x. Графически эволюция динамической системы во времени представляется движением точек в фазовом пространстве. Важным понятием теории диссипативных динамических систем является аттрактор — подмножество фазового пространства $A \subseteq M$, инвариантное относительно эволюции в системе:

$\psi^{(t)}(A) = A$ для всех $t \in T$ и такое, что существует окрестность U множества A, в которой для всех $y \in U$ выполняется равенство

$$\lim_{t \to \infty} \psi^{(t)}(y) = A.$$

Простые аттракторы — устойчивое состояние равновесия с неподвижной точкой x^*:

$$\lim_{t \to \infty} \psi^{(t)}(y) = x^*$$

и устойчивый цикл, отвечающий режиму периодических автоколебаний.

Множество точек, приводящих к некоторому аттрактору, называется его областью, или бассейном, притяжения — *basin of attraction*. Рассмотрим динамическую систему как полугруппу итераций $\{\psi^{(j)}\}_{j \geqslant 0}$, и пусть $R_0, R_1, R_2, \ldots$ — последовательность точек, описывающая эволюцию системы, определенная условием $R_{j+1} = \psi(R_j)$ при всех $j \geqslant 0$. Качественное поведение динамической системы с оператором эволюции в виде (1) зависит от параметра a, он является управляющим. До определенного значения a, не превышающего бифуркационное, система стремится к точечному аттрактору $R^* = \ln(a)/b$. Первый метаморфоз поведения системы происходит, когда производная, вычисленная в неподвижной точке, перестает удовлетворять критерию устойчивости. Для (1) это происходит при выполнении условия $a > \mathbf{e}^2$:

$$\psi'(R) = a\mathbf{e}^{-bR} - bRa\mathbf{e}^{-bR};$$

$$\psi'(R^*) = a\mathbf{e}^{-b\frac{\ln a}{b}} - b\frac{\ln a}{b}a\mathbf{e}^{-b\frac{\ln a}{b}} = \frac{a(1 - \ln a)}{\mathbf{e}^{\ln a}} = 1 - \ln a;$$

$$1 - \ln a = -1, \quad a = \mathbf{e}^2.$$

Теперь очевидно, что параметр b не влияет на топологические характеристики фазового портрета. Дина-

мическая система стремится в устойчивое циклическое состояние с периодом 2 — глобальный аттрактор, состоящий из двух периодических точек — последовательности (R_{T1}, R_{T2}, R_{T1}, R_{T2}...), областью притяжения которого является все фазовое пространство. Тот факт, что отображение Рикера имеет в этом диапазоне параметра цикл с периодом 2, говорит о том, что не стоит от построенного по эмпирическим данным графика ожидать характерной куполообразной кривой, как на рис. 1. Если далее увеличивать параметр a, будет увеличиваться амплитуда колебаний, и по достижению следующего порогового значения $a > 12{,}51$ произойдет бифуркация удвоения периода и установится цикл периода 4. При дальнейшем увеличении параметра a будет происходить каскад бифуркаций удвоения периода. При значении коэффициента $a > 14{,}8$ невозможно выделить устойчивые точки или замкнутый цикл — происходит детерминированный хаос, напоминающий стохастический процесс (на рис. 2 показана временная диаграмма), очень чувствительный к начальным условиям. Траектория притягивается к подмножеству фазового пространства, получившему вследствие своих геометрических свойств название "странный аттрактор". Как показали дальнейшие исследования, сценарием перехода к хаосу, описанным на примере квадратичного отображения $x_{n+1} = \lambda x_n(1 - x_n)$ Митчеллом Фейгенбаумом [4], формула (1) не ограничивается. Для некоторых диапазонов $a > 14{,}8$ наблюдаются "окна периодичности" — появляются устойчивые циклы нечетных периодов, т. е. для (1) возможны касательные бифуркации. В диапазоне значений управляющего параметра $18{,}474 < a < 18{,}564$ появление странного аттрактора не наблюдается, возникает устойчивый цикл.

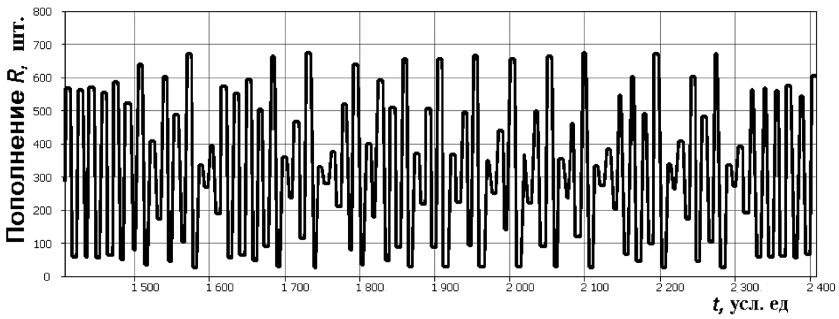


Рис. 2. Динамический хаос в модели Рикера

Многие нелинейные дискретные отображения обладают свойствами, делающими их объектом исследования теории динамического хаоса, например "аттрактор Эно". Ли и Йорк в 1975 г. опубликовали знаменитую статью "Period three implies chaos", в которой показали, что если одномерное отображение вида $R_{j+1} = \psi(R_j)$ при некотором значении одного из параметров имеет цикл периода 3, то оно также имеет и бесконечное множество циклов других периодов.

Для построения кривых запас—пополнение были предложены довольно сложные преобразования исходных данных наблюдений. Наверное, никогда исследователи не бывают так настойчивы, как в случае, когда пытаются отыскать то, чего нельзя обнаружить. Возникает естественный вопрос: имеют ли смысл методы построения кривой, если эмпирические данные о воспроизводстве, которые не находятся в стадии деградации популяций и для которых справедлива зависимость Рикера, будут представлены в виде сгущений точек на графике, мало напоминающих ожидаемую кривую. Однако изучение данных о деградирующей популяции позволили автору выявить некоторые важные особенности процесса формирования пополнения.

Общей целью всех проводимых автором с 2004 г. исследований является модельный анализ процессов в депрессивной экосистеме Каспия. Среди разнообразной информации, часто весьма недостоверной, привлекли внимание интересные сведения о численности молоди севрюги *Acipenser stellatus*. Численность пропущенных на нерест в Волгу производителей за период наблюдений изменялась очень существенно: от 230 тыс. в 1979—1981 гг. до 15 тыс. в 2000 г. [5].

Новая модель процесса формирования пополнения

Наличие ограниченных пищевых ресурсов в неявном виде учитывалось известными моделями, но общим недостатком существующих моделей (в том числе модели Шепарда и Бивертона—Холта) является неучет декомпенсационного фактора смертности. Не обращалось внимания и на изменения пищевых потребностей по мере развития молоди, которое происходит синхронно с началом сезонного уменьшения кормовой базы. Декомпенсационные факторы, увеличивающие смертность при уменьшении плотности, снижают эффективность нереста, уменьшая количество икры, реально вступившей в репродуктивный процесс при $S \ll R^*$. Введение такой зависимости становится очевидной необходимостью для моделирования популяций, подвергающихся "перелову".

Целью разработки новой модели запас—пополнение стало создание гибкого математического аппарата для задачи согласования характера поведения имитационной модели динамики популяции со статистическими данными о популяциях, так как очевидно, что поведение модели зависит от математических особенностей выбранной функции воспроизводства. Куполообразная кривая подразумевает наличие диапазона оптимальной численности нерестового стада.

Рассмотрим увеличение пищевых потребностей молоди и будем исходить из того факта, что темп роста находится в обратной зависимости от численности поколения, но не в обратно пропорциональной. Это согласуется с данными наблюдений биологов, в частности, с результатами экспериментов над ростом камбал при различной их плотности. Согласно наблюдениям, при увеличении плотности возникает асимметричное

распределение размерной структуры популяции в сторону преобладания особей с меньшими размерами. Описывать убыль численности поколения N на определенном интервале гибридного времени $[0, T]$ будут следующие объединенные в систему дифференциальные уравнения:

$$\begin{cases} \dfrac{dN}{dt} = -(\alpha w(t)N(t) + \theta(S)\beta)N(t); \\ \dfrac{dw}{dt} = \dfrac{gN^n(t)}{N^k(t)}, \quad k > n, \ \theta(S) = \dfrac{1}{1 - \exp(-cS)}; \\ S = \left.\dfrac{dN}{dt}\right|_{t=T}, \ \left.\dfrac{dN}{dt}\right|_{t=0} = \lambda S, \end{cases} \qquad (2)$$

где α, β, c — константы; S — нерестовый запас; $w(t)$ отражает изменение пищевых потребностей в зависимости от массы особей; g — ограничивающий фактор, учитывающий число доступных кормовых организмов; убывающая функция $\theta(S) \to 1$ при $S \to \infty$ отражает наличие эффекта Олли; λ — средняя плодовитость особей популяции.

Графиком в координатах S, $N(T)$ предложенной автором и исследованной в инструментальной среде разработки имитационных моделей AnyLogic5 новой модели является куполообразная кривая с уменьшающимся наклоном ниспадающей правой ветви (рис. 3) и имеющая две нетривиальные точки пересечения с биссектрисой координатного угла. Продукт AnyLogic предлагает достаточный выбор численных методов с изменяющимся шагом интегрирования для работы с системами обыкновенных дифференциальных уравнений в форме Коши и дифференциальных уравнений с отклоняющимся аргументом. Это средство, имеющее и ряд недостатков, может служить до некоторой степени малобюджетной альтернативой системе MatLab.

Поведение траектории динамической системы, использующей в качестве оператора эволюции модель (2), качественно отличается от системы на основе формулы Рикера возможностью притяжения к двум аттракторам и, соответственно, наличием двух областей притяжения, границей между которыми служит репеллер — неустойчивая особая точка первого пересечения кривой с биссектрисой координатного угла. Траектории с начальными условиями, разделенными репеллером, покидают его окрестность и приближаются к разным аттракторам. Один из аттракторов — точка с координатами (0, 0) на плоскости $R \times S$. Если начальная численность популяции соответствует области притяжения этого аттрактора, произойдет вымирание популяции.

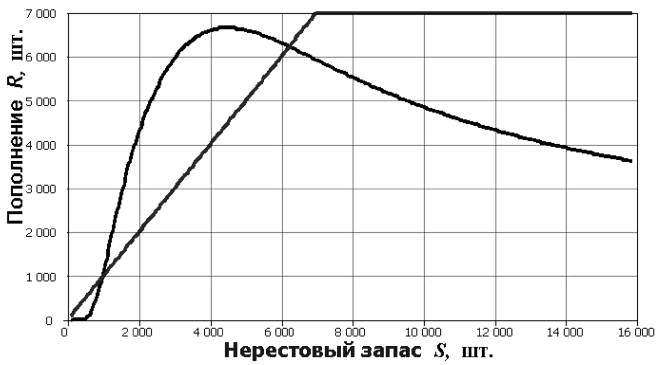


Рис. 3. Кривая запас—пополнение модели (2)

Другое отличие состоит в том, что в динамической системе на основе системы (2) нет бифуркационных значений коэффициентов. Она является структурно устойчивой (грубой по терминологии Понтрягина), топологическое поведение траекторий динамической системы сохраняется при изменениях параметров.

Об оптимизации промысла с учетом предложенной модели

Цели промысла всегда состоят в достижении максимально возможного уровня эксплуатации MSY (*maximum sustainable yield*) — наибольшего среднего улова, который можно на протяжении длительного времени изымать из популяции. Определение допустимого улова — непростая задача с несколькими неизвестными, решаемая обычно с помощью экспертных оценок. Большинство экспертов мыслят в терминах линейной причинности, но сложные системы подчас обладают свойствами, вводящими в заблуждение наблюдателя. Часто оказывается, что долгосрочная реакция системы на воздействие оказывается прямо противоположной краткосрочной. Для такого характера реакции Дж. Форрестер предложил термин "антиинтуитивный" (противоречащий "здравому смыслу") тип реакции на внешнее воздействие. Подобное непредвиденное поведение не раз приводило к провалу выбранной стратегии управления экосистемой.

Проблема прогнозирования реакции природных систем на антропогенное вмешательство еще очень долго будет оставаться трудноразрешимой. В 1975 г. был опубликован [6] прогноз, согласно которому выпуск 90 млн шт. молоди осетровых позволит довести уловы до 30 тыс т. Еще ранее декларировалась задача по превращению Каспия в "осетрово-килечный" водоем [7]. Как итог рыбохозяйственной деятельности, с середины 1990-х годов констатируются деградация популяций осетровых, приведшая (с фатальным опозданием!) к прекращению промысла, и падение уловов кильки в 2000 г. Требуемых масштабов выпуска молоди рыбоводным заводам в середине 1980-х годов достичь удалось, а стабильного улова нет. Массовый одноместный выпуск молоди рыбоводными заводами приводит к неестественной для ее среды концентрации, увеличивает внутривидовую конкуренцию и плотностно-зависимую смертность. Такой эффект, собственно, и прогнозируют нелинейные модели запас—пополнение.

Рассмотрим популяцию как непрерывно-дискретную динамическую систему с учетом воздействия промыслового изъятия. Пусть некоторая доля q нерестового запаса изымается промыслом, тогда начальные ус-

ловия первого уравнения системы (2) будут определяться следующим образом:

$$\left.\frac{dN_{i+1}}{dt}\right|_{t=0} = \lambda(1-q)\left.\frac{dN_i}{dt}\right|_{t=\tau}, \quad 0 \leqslant q \leqslant 1.$$

Тогда популяция может существовать неограниченно долгое время, не вымирая при условии, что не попадет в область притяжения аттрактора $R^{(0)}$. В случае соблюдения условия стабилизация популяции, в которой происходит изъятие некоторой процентной доли особей, произойдет в новой неподвижной точке $R^{**} = \psi((1-q)R^{**})$.

Особый интерес представляет случай, когда величина q совсем незначительно превосходит максимально допустимое промысловое изъятие. Вылов при изменении режима промысла увеличивается (определяющие ОДУ "эксперты" могут впасть в эйфорию), затем медленно падает (рис. 4), потом у наблюдателя создается иллюзия стабилизации запаса. В модельном эксперименте размер вылова через девять лет вернулся к уровню, получаемому до увеличения доли промыслового изъятия, а от момента усиления промыслового давления до полного вымирания популяции прошло 57 лет.

Подобная динамика наблюдалась после увеличения уловов каспийских осетровых в конце 70-х годов. Еще в 1986 г. Р. П. Ходоревской оптимистично прогнозировалось незначительное увеличение уловов севрюги и белуги [8]. Экологи начали "бить тревогу" и писать о деградации запасов в начале 90-х годов.

Максимальный устойчивый улов MSY можно получать неограниченно долгое время, когда популяция находится в состоянии, продуцирующем максимально эффективное воспроизводство: $(R_{n+1} - R_n) \to \max$. В таком случае промысел будет забирать весь излишний прирост численности. Число выловленных особей составит $Y = \psi(R_{\text{опт}}) - R_{\text{опт}}$. Задача оптимизации состоит в выборе такого коэффициента изъятия, который бы стабилизировал популяцию в состоянии $R_{\text{опт}}$:

$$\forall n R_n > R_{\text{опт}}, \lim_{n\to\infty}(1-q)\psi^n(R_0) = R_{\text{опт}}.$$

В случае, если текущая численность популяции $R_n < R_{\text{опт}}$ то промысел должен быть **запрещен** до тех пор, пока популяция не достигнет оптимального для промысла состояния. Но такой уровень запаса можно определить только экспериментально по эмпирическим данным, а подобные эксперименты связаны с риском деградации популяции. Подобное "оптимизированное" состояние популяции будет неустойчиво, случайные изменения численности или перелов могут подвести к границе, после которой устойчивым останется только тривиальное состояние равновесия. Как справедливо заметил чешский гидробиолог М. Страшкраба,

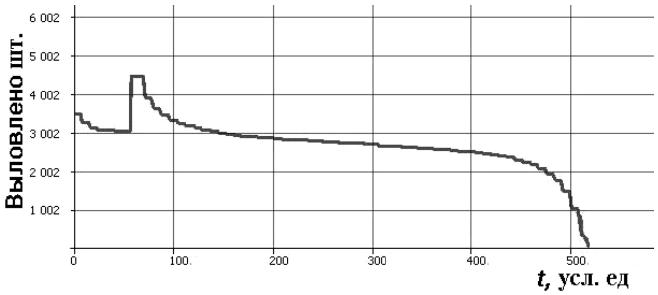


Рис. 4. Динамика улова деградирующей популяции на основе модели (2)

применение принципов оптимальности в экосистеме осложняется трудностями фундаментального порядка, обусловленными невозможностью существования одной общей цели для такой гетерогенной структуры.

Причиной деградации запасов севрюги, сохранившей большую часть нерестилищ после зарегулирования Волги, стало превышение допустимого изъятия из нерестовой части популяции в конце 1970-х и первой половине 1980-х годов и не прекращение промысла в середине 1990-х годов. Подобная картина наблюдалась ранее с популяциями реки Кура и с другим когда-то промысловым видом — осетром *Acipencer sturio*, промысел которого велся в реках северной Европы, где сегодня он больше не встречается. Хорошо документирован коллапс без последующего восстановления промысловых запасов форели и сельди Великих Озер в 1950-е годы.

Все, кто пытается добиться максимальной прибыли от экосистемы исходя из "принципов оптимальности", забывают о так называемом "принципе хрупкости хорошего".

Список литературы

1. **Пригожин И.** Философия нестабильности // Вопросы философии. 1991. № 6. С. 46—57.

2. **Шибаев С. В.** Промысловая ихтиология. СПб.: Проспект науки, 2007. 400 с.

3. **Ricker W.** Stock and recruitment // Journal Fisheries research board of Canada. 1954. № 11. С. 559—623.

4. **Фейгенбаум М.** Универсальность в поведении нелинейных систем // Успехи физических наук. 1983. Т. 141. Вып. 2. С. 343—374.

5. **Довгопол Г. Ф., Вещев П. В.** Оценка численности поколений севрюги Acipenser stellatus и основных факторов, влияющих на структуру ее популяции // Вопросы ихтиологии. 1993. Т. 33. С. 93—99.

6. **Мильштейн В. В.** Перспективы воспроизводства осетровых в Каспийском бассейне // Биологическая продуктивность Каспийского моря. М.: Пищевая промышленность, 1975. С. 209—213.

7. **Шорыгин А. А.** Питание и пищевые взаимоотношения рыб Каспийского моря. М.: Пищепромиздат, 1952. 267 с.

8. **Динамика** численности промысловых рыб. М.: Наука, 1986. С. 189—199.

УДК 654.9.001.25

Р. А. Дурнев, канд. техн. наук, ст. науч. сотр.,
Центр стратегических исследований
гражданской защиты МЧС России

Система информирования и оповещения населения: функции и структура

Рассматривается новая система информирования и оповещения населения, основанная на современных информационно-телекоммуникационных технологиях. Определены ее роль и место в общей системе обеспечения безопасности жизнедеятельности, установлены основные функции и структура.

Ключевые слова: *информирование и оповещение населения, информационно-телекоммуникационные технологии, информационный центр, терминальный комплекс.*

Анализ состояния информирования и оповещения населения

Ежегодно в Российской Федерации в чрезвычайных ситуациях (ЧС), дорожно-транспортных происшествиях (ДТП), при пожарах и авариях на водных объектах (далее — опасных и чрезвычайных ситуациях) погибает свыше 65 тыс. человек, получает травмы около 300 тыс. человек, прямой материальный ущерб составляет более 100 млрд руб. [1, 2].

Практика показывает, что большую роль в снижении людских потерь и материального ущерба в указанных ситуациях играют информирование и оповещение населения. От регулярности предоставления сведений о возможных источниках и масштабах опасных и чрезвычайных ситуаций, мерах по уменьшению их последствий, а также от оперативности доведения сигнала оповещения до сил предупреждения и ликвидации ЧС и до населения зависит, в конечном итоге, результативность укрытия людей в защитных сооружениях, их эвакуации из зоны ЧС и других способов защиты.

Анализ нормативных правовых актов [3—5] показывает, что информирование и оповещение занимают важное место в системе мероприятий по защите населения и территорий от опасных и чрезвычайных ситуаций (рис. 1), а создание и поддержание в постоянной готовности соответствующих систем является важнейшей задачей органов государственной власти, местного самоуправления и организаций.

Для информирования и оповещения населения используются ресурсы средств массовой информации, созданы и функционируют системы централизованного и локального оповещения. Однако анализ показывает, что существующие технические средства информирования и оповещения (ТСИО) морально и физически устарели, имеют низкие тактико-технические характеристики. Более половины региональных систем централизованного оповещения превысили установленные сроки эксплуатации. Ежегодная стоимость их содержания и обслуживания превышает остаточную (балансовую) стоимость таких систем. Число локальных систем оповещения, создаваемых в районах размещения потенциально опасных объектов, составляет менее 35 % от потребности.

В целом, существующие системы информирования и оповещения охватывают менее 45 % населения страны. Принимая во внимание прогнозную оценку числа неработоспособных ТСИО к 2010 г., охват населения мероприятиями оповещения и информирования может составить менее 15 %.

Все это свидетельствует об острой потребности в реконструкции и модернизации существующих систем информирования и оповещения. Однако, как показывает анализ, в настоящее время отсутствуют соответствующие концептуальные и программные документы. В ряде нормативных правовых актов, регламентирующих функционирование этих систем [6—8], приводится перечень лишь отдельных мер, направленных на решение частных вопросов поддержания в готовности ТСИО, другие, например [9], практически не реализуются. В связи с этим особенно злободневной является проблема дальнейшего применения рассматриваемых способов защиты населения.

Требования к новой системе информирования и оповещения населения

Для решения указанной проблемы представляется целесообразным создание новой системы, которая бы обеспечивала [10]:

- осуществление функций оповещения, информирования и подготовки, формирования культуры безопасности жизнедеятельности (КБЖ) населения;

Рис. 1. Место информирования и оповещения в общей системе мероприятий по защите населения и территорий от опасных и чрезвычайных ситуаций

- максимально полный и оперативный охват населения независимо от его местонахождения;
- обратную связь с местами пребывания оповещаемых и информируемых людей;
- комплексное использование цифровых технологий связи и вещания, средств сотовой связи, электронно-вычислительной техники, Интернет-ресурсов;
- высокую надежность и живучесть в условиях воздействия поражающих факторов источников ЧС мирного и военного времени;
- полное сопряжение с аппаратно-программными комплексами органов управления гражданской обороны (ГО) и единой государственной системы предупреждения и ликвидации чрезвычайных ситуаций (РСЧС);
- самоокупаемость за счет использования части информационного ресурса в коммерческих целях и др.

Очевидно, что реализация указанных положений невозможна без использования современных информационно-телекоммуникационных технологий, под которыми понимаются методы и средства сбора, обработки, хранения, передачи, приема и отображения аудиовизуальной информации [10].

Рис. 2. Место систем информирования и оповещения населения в системе обеспечения безопасности личности, общества и государства

В наше время именно эти технологии определяют облик не только экономически развитых стран, но и всего мирового сообщества. Поэтому современную ступень развития цивилизации принято характеризовать как информационное общество. Основными его чертами является увеличение роли информации и знаний, доли информационных коммуникаций, продуктов и услуг в валовом внутреннем продукте, создание глобального информационного пространства, обеспечивающего эффективное взаимодействие людей, их доступ к мировым информационным ресурсам и удовлетворение их социальных и личных потребностей.

Для применения этих технологий в интересах защиты населения в настоящее время создается Общероссийская комплексная система информирования и оповещения населения в местах массового пребывания людей (ОКСИОН).

Функции и структура новой системы информирования и оповещения населения

Для установления функций и состава элементов ОКСИОН определялись ее роль и место в общей системе обеспечения безопасности. Для этого в соответствии с положениями нормативных правовых актов [3—5] и схемы на рис. 1 установлены основные системы по обеспечению безопасности личности, общества и государства [10]. На основании этого выявлено место систем информирования и оповещения населения в общей системе обеспечения безопасности страны (рис. 2).

Из данного рисунка видно, что система гражданской защиты включает три подсистемы — систему управления гражданской защитой, систему сил и средств гражданской защиты и систему подготовки, информирования и оповещения населения. Органичной частью последней подсистемы и является ОКСИОН.

Для установления роли систем информирования и оповещения населения необходимо определить цели их функционирования. Для этого построено соответствующее дерево целей, фрагмент которого представлен на рис. 3. Цели Ц2.2.3.2, Ц2.2.3.3, Ц2.2.3.4 и Ц2.2.3.5, входящие в Ц2.2.3, являются основными задачами ОКСИОН и позволяют сформулировать следующую основную цель ее функционирования: информирование и оповещение населения с использованием информационно-телекоммуникационных технологий (ИТТ) для повышения эффективности его действии при угрозе и возникновении ЧС мирного и военного времени.

Для достижения указанной цели определено, что основными функциями системы должны являться:

- оповещение населения, реализуемое при угрозе опасных и чрезвычайных ситуаций, в рамках которой используются звуковые сигналы оповещения, а также краткая звуковая или текстовая информация по порядку действий;
- информирование населения, в ходе которого при угрозе и развитии опасных и чрезвычайных ситуаций транслируется аудиовизуальная информация по правилам поведения в зоне этих ситуаций, местам нахождения медпунктов, пунктов жизнеобеспечения, телефонам горячих линий;

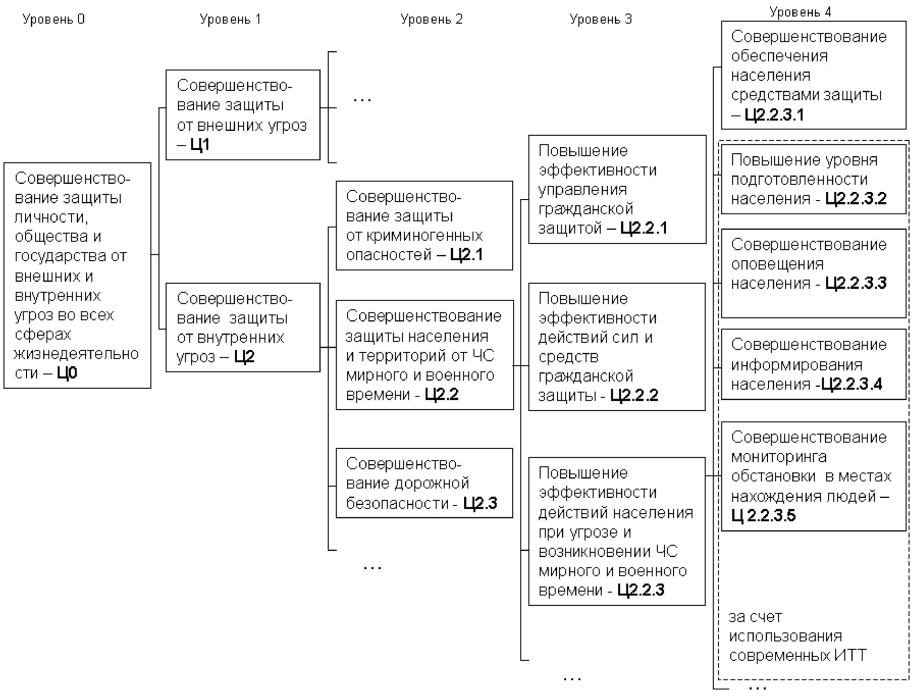


Рис. 3. Фрагмент дерева целей по совершенствованию защиты личности, общества и государства от внешних и внутренних угроз во всех сферах жизнедеятельности

- подготовка населения, в рамках которой в повседневном режиме населению транслируются видео- и анимационные ролики, направленные на формирование норм и ценностей безопасного поведения, КБЖ;
- мониторинг обстановки в местах массового пребывания людей, реализуемый во всех рассматриваемых периодах.

Структура ОКСИОН представлена на рис. 4.

Информационные центры предназначены для: планирования и проведения информационных операций; управления трансляциями на терминальных комплексах в зоне ответственности и функционированием нижестоящих информационных центров; анализа информации об обстановке в местах массового пребывания людей; контроля работоспособности функционирования терминальных комплексов; организации взаимодействия с территориальными Центрами управления в кризисных ситуациях (далее — ЦУКС), системами информирования и оповещения населения другой ведомственной принадлежности и иных форм собственности.

Терминальные комплексы предназначены для приема, обработки и отображения аудиовизуальных сообщений, а также приема и передачи в информационные центры информации об обстановке в местах массового пребывания людей. Они разделяются на стационарные терминальные комплексы (СТК) и мобильные терминальные комплексы (МТК).

СТК включают технические средства сбора и отображения информации, радиационного и химического контроля, звукового вещания. К техническим средствам сбора информации относятся обзорные видеокамеры, позволяющие фиксировать и передавать информацию об обстановке в местах расположения терминальных комплексов, которые находятся на наиболее потенциально опасных направлениях в местах массового пребывания людей, а также вызывные голосовые панели для связи информационных центров с операторами. К средствам радиационного и химического контроля относятся автоматизированные комплексы, включающие датчики, блоки детектирования, коммутирующие устройства, блоки сбора и хранения данных по радиационной и химической обстановке в местах массового пребывания людей. Средства звукового вещания включают устройства усиления звука, динамики и другое оборудование, необходимое для звукового оповещения населения. Технические средства отображения информации включают уличные светодиодные панели, плазменные экраны внутри зданий, экраны "бегущая строка".

По местам установки и составу оборудования СТК подразделяются на пункты уличного информирования и оповещения населения (ПУОН) и пункты информирования и оповещения в зданиях с массовым пребыванием людей (ПИОН). ПУОН располагаются вне зданий и включают светодиодный экран, камеры видеонаблюдения, звукоусиливающее оборудование, оборудование для радиационного и химического контроля и др.

ПИОН располагают в зданиях с массовым пребыванием людей и включают полноцветный экран (плазменный) или устройство бегущей строки, камеры видеонаб-

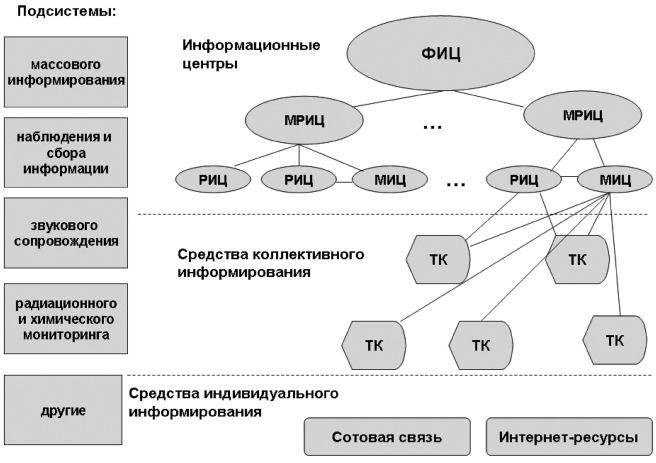


Рис. 4. Структура ОКСИОН
ФИЦ — федеральный информационный центр; МРИЦ — межрегиональные информационные центры; РИЦ — региональные информационные центры; МИЦ — местные (муниципальные) информационные центры; ТК — терминальные комплексы

людения, звукоусиливающее оборудование, оборудование для радиационного и химического контроля и др.

МТК включают транспортные средства, на которых размещаются светодиодные экраны с оборудованием, необходимым для отображения видео- и аудиоинформации, видеонаблюдения, обеспечения связи, создания информационного контента, а также мониторинга радиационной, химической и биологической обстановки, автономного энергоснабжения, защиты от поражающих факторов источников ЧС и другим оборудованием.

Распределенные автоматизированные подсистемы предназначены для обеспечения сопряжения между информационными центрами и терминальными комплексами и включают подсистемы массового информирования, наблюдения и сбора информации, связи и передачи данных, информационной безопасности, радиационного и химического контроля, звукового сопровождения и информирования, контроля и управления ОКСИОН, часофикации, геоинформационную подсистему.

Возможности новой системы информирования и оповещения населения

Развертывание ОКСИОН на территории субъектов Российской Федерации и муниципальных образований в полном масштабе планируется поэтапно к 2010 г. Это позволит более чем в 3 раза увеличить охват населения мероприятиями по гарантированному оповещению и оперативному информированию об угрозе и возникновении ЧС и террористических акциях. Затраты бюджетных средств на ликвидацию чрезвычайных ситуаций и последствий террористических акций уменьшатся в 3,4 раза [10].

В области гражданской обороны в особый период (при переводе ГО с мирного на военное положение) будет обеспечена непрерывность управления ГО, поступления информации, сигналов оповещения и т. п.

В правоохранительной области эффект от создания и функционирования ОКСИОН будет достигаться за счет повышения действенности мониторинга за общественным порядком в местах массового пребывания людей, увеличения результативности процессов обнаружения и идентификации социально опасных лиц.

Кроме того, развертывание ОКСИОН будет способствовать развитию науки, передовых наукоемких информационных технологий, промышленности, систем связи и телекоммуникации, созданию новых рабочих мест.

Таким образом, создание ОКСИОН в местах массового пребывания людей будет значимо способствовать формированию культуры безопасности жизнедеятельности, повысит эффективность мероприятий оповещения и информирования населения и явится одним из факторов обеспечения стабильного социально-экономического развития регионов страны и России в целом.

Список литературы

1. **Государственный** доклад о состоянии защиты населения и территорий Российской Федерации от чрезвычайных ситуаций природного и техногенного характера в 2006 году. М.: ФГУ ВНИИ ГОЧС (ФЦ), 2007. 204 с.
2. **Дорожно-транспортные** происшествия в России (2005 г.): Информационно-аналитический сборник. М.: ДОБДД МВД России, 2006. 106 с.
3. **О защите** населения и территорий от чрезвычайных ситуаций природного и техногенного характера: Федеральный Закон от 21 декабря 1994 г. № 68-ФЗ.
4. **О гражданской** обороне: Федеральный закон от 12 февраля 1998 г. № 28-ФЗ.
5. **О пожарной** безопасности: Федеральный закон от 21 декабря 1994 г. № 69-ФЗ.
6. **Об утверждении** Положения по организации эксплуатационно-технического обслуживания систем оповещения населения: Приказ МЧС России, Минсвязи России, Минкультуры России № 877/138/597 от 7 декабря 2005 года.
7. **Об утверждении** Положения о приоритетном использовании, а также приостановлении или ограничении использования любых сетей связи и средств связи во время чрезвычайных ситуаций природного и техногенного характера: Постановление Правительства Российской Федерации от 31 декабря 2004 г. № 895.
8. **Об утверждении** Концепции информационной безопасности МЧС России: Приказ МЧС России от 07.03.2007 г. № 121.
9. **Программа** реконструкции систем оповещения гражданской обороны Российской Федерации до 2010 года. Утверждена приказом МЧС России от 10.10.2000. № 508.
10. **Дурнев Р. А.** Информирование и оповещение населения: роль и место в системе обеспечения безопасности жизнедеятельности: Сб. трудов ЦСИ ГЗ МЧС России. М.: ЦСИ ГЗ МЧС России, 2007. Вып. 33.

Новые книги

Алешин Л. И. Информационные технологии. Учеб. пос. М.: Маркет ДС, 2008. 384 с.

Учебное пособие посвящено проблеме современного информационного общества — широкомасштабному применению информационных технологий в различных отраслях и предметных областях. Эта проблема актуальна практически для всех информационных и иных заведений, подразделений, организаций, предприятий и офисов. Современному человеку необходимо знать информационные технологии, уметь успешно применять данные знания при решении как личностных, так и производственных задач повседневной жизни. В пособии рассматриваются основные теоретические и практические аспекты проблемы, а также терминологический аппарат и другие сведения, связанные с информационными технологиями.

Издание предназначено для студентов информационных и иных специальностей и специализаций, изучающих вопросы, связанные с информационными технологиями и их применением в различных предметных областях. Оно также представляет интерес для аспирантов, преподавателей, информационных работников, ИТ-специалистов и всех увлеченных новыми информационными технологиями и возможностями их применения.

CONTENTS

technology, is offered. The advantages of the usage of the developed model of the storage of objects are formulated.
Keywords: the information storage, the quasistructured data, the automated information systems.

The new approach to aggregation of a multidimensional data sets is offered. It is based on use of uniformity's criterion determined as a maximum of a uniformity's parameter of an one-dimensional file. The parameter represents the relation of average harmonious values of a file elements to average arithmetic. Generalization of this definition on a matrixes, as average harmonious of uniformity's parameters of her columns is resulted. Properties of a uniformity's parameter, major of which — invariancy to transformations of similarity and inversion are considered. These properties allow to create effective algorithms of aggregation of the multidimensional data sets. Examples of use of the offered criterion in the theory of decision-making, for construction of an integrated parameter of reliability of banks and gradation of attributes for their forecasting are considered. Researchers are maintained by the Russian Federal Property Fund (the grant № 05-08-65501).
Keywords: uniformity's criterion, parameter, matrix, transformation of inversion, gradation of attributes.

Author suggests a new discrete-continuous model for estimation and forecasting of efficiency of reproduction of food fish population. Model based on threshold effects in early stage of fish ontogenesis, which investigate within the bounds of stage-development theory. Phase portrait of the dynamic system within suggested model has qualitative sense differences from the Ricker Map or Beaverton-Holt Map. Author draws a conclusion about dangers concerned with transferring population to the point of optimum yield. Article assigned for specialists in field of implementation of information technologies and simulation models in ecology.
Keywords: simulation of population dynamic, nonlinear dynamic systems, optimization of natural resources.

It is considered new system informing and notifications of the population, founded on modern information-telecommunication technology. They are determined its role and place in the general system of the provision to safety to vital activity, are installed main functions and structure.
Keywords: leaving out and warning of the population, information-telecommunication technology, documentation centre, terminal complex.

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. Пчёлкина*

Сдано в набор 06.11.2008. Подписано в печать 11.12.2008. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л.10,78. Уч.-изд. л. 12,30. Заказ 19. 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