

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

5(141)
2008

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

СЕТИ И СИСТЕМЫ СВЯЗИ

Аджалов В. И., Тарасов В. Ю. Методы обслуживания голосовых вызовов в сети Интернет 2
Шахов В. Г., Нопин С. В. Использование IP-телефонии в защищенном режиме 7

WEB-ТЕХНОЛОГИИ

Свечников С. В. Высокореlevantный поиск и автоматическая категоризация
ресурсов сети Интернет 11
Чеснаковский А. А. Семантическое отслеживание изменений на веб-сайтах 16

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ

Жусов Д. Л. Алгоритм обнаружения компьютерных атак на web-сервер. 23
Мистров Л. Е. Метод оценки эффективности применения комплексов информаци-
онной индивидуальной и групповой безопасности организационно-технических
систем в конфликтной неопределенности 26

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ И КОМПЬЮТЕРНАЯ ГРАФИКА

Агеев Л. Д., Калашников В. С. Компьютерное проектирование проводящего рисунка
плоских квазилогопериодических антенн 31
Плоткин Д. А. Новый метод сжатия изображений, построенный на основе JPEG
Baseline и метода бинарных интервальных преобразований. 34

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

Пекунов В. В. Автоматизация параллельного программирования при моделирова-
нии многофазных сред 37
Черемисинов Д. И. Разработка параллельных программ для решения логико-комби-
наторных задач 42
Князев Е. Г., Шопырин Д. Г. Автоматизированная классификация изменений про-
граммного кода методами многомерного статистического анализа. 48

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Кулаков С. М., Трофимов В. Б., Бондарь Н. Ф., Чабан С. В. Интеллектуальная сис-
тема распознавания поверхностных дефектов проката 53
Алгулиев Р. М., Абдуллаев Р. С. Анализ способов сокращения транспортных нагру-
зок на городскую инфраструктуру 59

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

Муханов Л. Е. Система обнаружения мошенничества в области платежных карт 62
Дрогин А. В. Разработка научно-методического и программного обеспечения под-
системы бюджетирования АСУП 66
Синицын С. В., Хлытчиев О. И. От процессов предприятия к основным объектам
системы документооборота программных проектов 70

ДИСКУССИОННЫЙ КЛУБ

Фридланд А. Я. О сущности информации: два подхода 75
Contents 86
Приложение. Батищев Д. И., Старостин Н. В., Филимонов А. В. Многоуровневая де-
композиция гиперграфовых структур

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

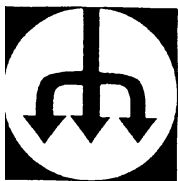
АВДОШИН С. М.
АНТОНОВ Б. И.
БАТИЩЕВ Д. И.
БАРСКИЙ А. Б.
БОЖКО А. Н.
ВАСЕНИН В. А.
ГАЛУШКИН А. И.
ГЛОРИОЗОВ Е. Л.
ГОРБАТОВ В. А.
ДОМРАЧЕВ В. Г.
ЗАЛЕЩАНСКИЙ Б. Д.
ЗАРУБИН В. С.
ИВАННИКОВ А. Д.
ИСАЕНКО Р. О.
КОЛИН К. К.
КУЛАГИН В. П.
КУРЕЙЧИК В. М.
ЛЬВОВИЧ Я. Е.
МАЛЬЦЕВ П. П.
МЕДВЕДЕВ Н. В.
МИХАЙЛОВ Б.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

Редакция:

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

Аннотации статей размещены на сайте журнала 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 004.738.5:621.395

В. И. Аджалов, д-р техн. наук, проф.,

В. Ю. Тарасов, канд. техн. наук,

valery.tarasov@mail.ru,

Московский физико-технический институт

Методы обслуживания голосовых вызовов в сети Интернет

Исследованы принципы централизованной обработки голосовых вызовов абонентов, поступающих со страниц Интернет-сайтов. Показано, что на основе анализа параметров вызова осуществимо соединение абонента с оператором определенной квалификации. Проанализирован алгоритм обслуживания вызовов в зависимости от территориальной и языковой принадлежности абонента, наличия предыдущих обращений, тематической направленности запроса. Рассмотрен способ предоставления информации о состоянии голосового обслуживания определенного Интернет-ресурса без предварительного инициирования вызова абонентом.

В настоящее время сфера информационных и коммуникационных технологий становится все более важной стратегической составляющей социально-экономического потенциала России. Эта отрасль развивается быстрее российской экономики в целом, демонстрируя высокие темпы роста на уровне около 20 % в год. Общий объем отрасли в 2005 году приблизился к 1 трлн руб., в том числе объем сектора информационных технологий превысил 300 млрд руб. Вклад услуг аутсорсинга обработки вызовов в объемы отрасли в 2005 году составил около 1 млрд руб.

В 2005 году число пользователей сети Интернет составило 22 млн чел. — это каждый восьмой житель России, а общее число эксплуатируемых компьютеров превысило 17 млн шт., увеличившись по сравнению с 2004 годом более чем на 16 %.

Используя Интернет, можно обмениваться цифровой информацией (наиболее известный пример — электронная почта). Однако технически возможным представляется оцифровать звук и передать его в цифровом виде по сети. По такому принципу построена работа программных приложений, использующих сеть Интернет для обмена голосовой информацией в режиме реального времени между двумя абонентами.

В данной статье авторы рассматривают обслуживание абонентов в сети Интернет и предлагают новые методы, позволяющие повысить эффективность централизованной обработки голосовых вызовов в сети передачи данных.

Назначение и принципы функционирования центров обработки вызовов

Центры обработки вызовов (ЦОВ, Call-центр, контакт-центр) обеспечивают развитие отрасли информационных и коммуникационных технологий в части автоматизации взаимодействия предприятий с клиентами. ЦОВ предназначен для эффективного приема большого числа телефонных вызовов и их обработки в соответствии с заданными бизнес-правилами. Возможности ЦОВ оптимизируют обслуживание абонентов за счет обработки входящих и исходящих вызовов, информационного самообслуживания абонентов, обработки факсов, писем электронной почты и web-вызовов, интеграции с корпоративными информационными системами заказчика и т. п.

Совокупность возможностей передачи голоса в сетях с пакетной коммутацией и методов централизованной обработки вызовов абонентов открывает широкие перспективы для предоставления новых видов голосовых услуг в сети передачи данных.

Большинство известных решений данной задачи нацелено на установление голосового соединения одного абонента с другим и не предназначено для централизованного обслуживания вызовов множества абонентов. Однако при поступлении голосового вызова из сети Интернет принимающей стороне становятся доступны данные, критичные для определения правил обслуживания вызывающего абонента. К данной категории параметров относятся:

- IP-адрес абонента (подлинный адрес компьютера абонента, либо адрес шлюза оператора связи);
- URL-адрес страницы, с которой инициирован вызов;
- статистические параметры Интернет-сервера (частота посещений абонентом данной страницы/сервера, время проведенное на странице и т. д.);
- параметры, введенные абонентом перед инициированием голосового вызова (к примеру, желание общаться с менеджером или специалистом);

- наличие регистрационных и идентификационных данных абонента и предыдущих обращений;
- данные, определяющие временные параметры передачи голосовых пакетов в IP-сети.

Значение IP-адреса абонента позволяет установить принадлежность абонента к той или иной территориальной зоне, что повышает качество обслуживания абонентов за счет сегментации вызовов по территориальному признаку. Определение URL-адреса страницы, с которой совершен вызов, позволяет распределить вызовы абонентов на обслуживание операторам, владеющим необходимыми профессиональными знаниями. Идентификация абонентов позволяет предоставить персонализированное обслуживание с различными уровнями качества сервиса.

Архитектура ЦОВ в сети Интернет

Предложенная архитектура центра обработки вызовов в сети Интернет с указанием маршрутов информационных потоков приведена на рис. 1. Для обмена голосовой информацией абонент использует компьютер с мультимедийными возможностями. Интернет-сайты клиентов представляют собой обслуживаемые в ЦОВ ресурсы сети Интернет. Страницы обслуживаемых сайтов содержат программный код, обеспечивающий установление голосового соединения абонента с оператором ЦОВ, а также отображение статистической информации о состоянии обслуживания вызовов (далее табло). Сервер ЦОВ содержит базу данных абонентов, операторов, таблицы и приложения, которые используют при обработке вызовов.

Поясним технологию обслуживания вызовов в сети Интернет. Посетитель страницы Интернет-сайта, обслуживаемого в ЦОВ, активирует на компьютере программный код, полученный со страницы данного сайта. При загрузке или обновлении содержания Интернет-страницы происходит выполнение программного кода, в результате которого посетителю на табло отображается информация о состоянии обслуживания абонентов данной категории (число операторов, размер очереди и т. д.).

При инициировании вызова, программный код передает серверу ЦОВ: URL-адрес страницы совершения вызова, IP-адрес абонента, e-mail адрес абонента (являющийся персональным кодом). Проанализировав полученные данные (ал-

горитм анализа представлен на рис. 2 и описан ниже), сервер ЦОВ определяет наличие свободного оператора требуемой квалификации, способного обслужить вызов на родном языке абонента.

Если необходимый оператор найден и свободен — сервер устанавливает двухстороннее голосовое соединение абонента с оператором через сеть Интернет, при этом на компьютере оператора отображают страницу, с которой абонент совершил вызов, а также персональную информацию абонента, если он идентифицирован по предыдущим обращениям. В случае, если требуемый оператор отсутствует или занят, вызов помещают в очередь, а на табло отображается размер очереди и предлагается форма заказа обратного вызова абонента оператором. В качестве размера очереди может отображаться порядковый номер абонента в очереди и/или расчетное время ожидания ответа оператора.

Распределение вызовов между операторами

На рис. 2 предложена обобщенная алгоритмическая схема обработки информации при обслуживании вызовов в сети Интернет. В ходе работы алгоритма сервер ЦОВ анализирует URL-адрес страницы совершения вызова, IP-адрес абонента, персональный идентификационный код абонента. По содержанию URL-адреса определяют тематику запроса абонента и предъявляют требования к квалификации обслуживающего оператора. По IP-адресу определяют территориальную и вероятную языковую принадлежность абонента. Персональный идентификатор абонента используют для определения предыдущих обращений абонента.

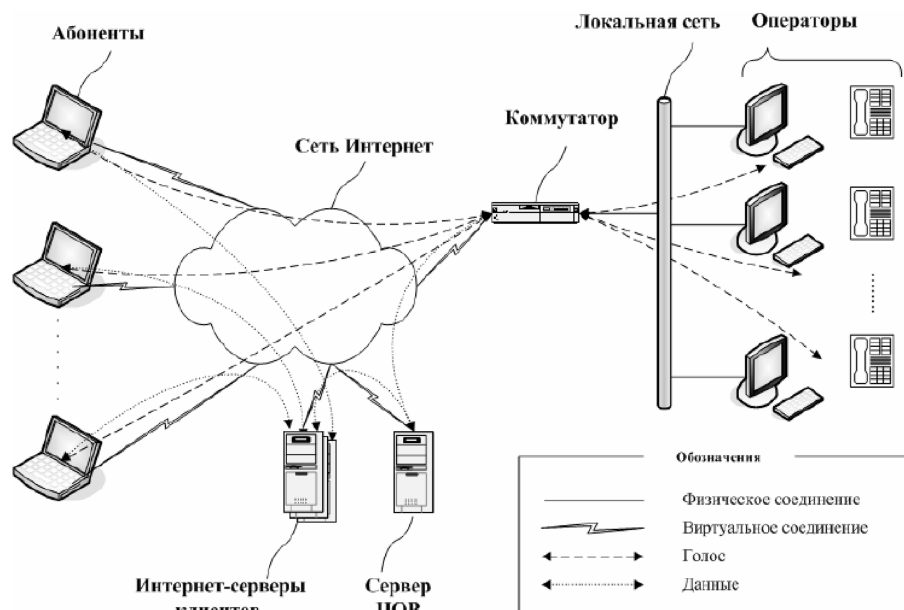


Рис. 1. Архитектура центра обработки вызовов в сети Интернет

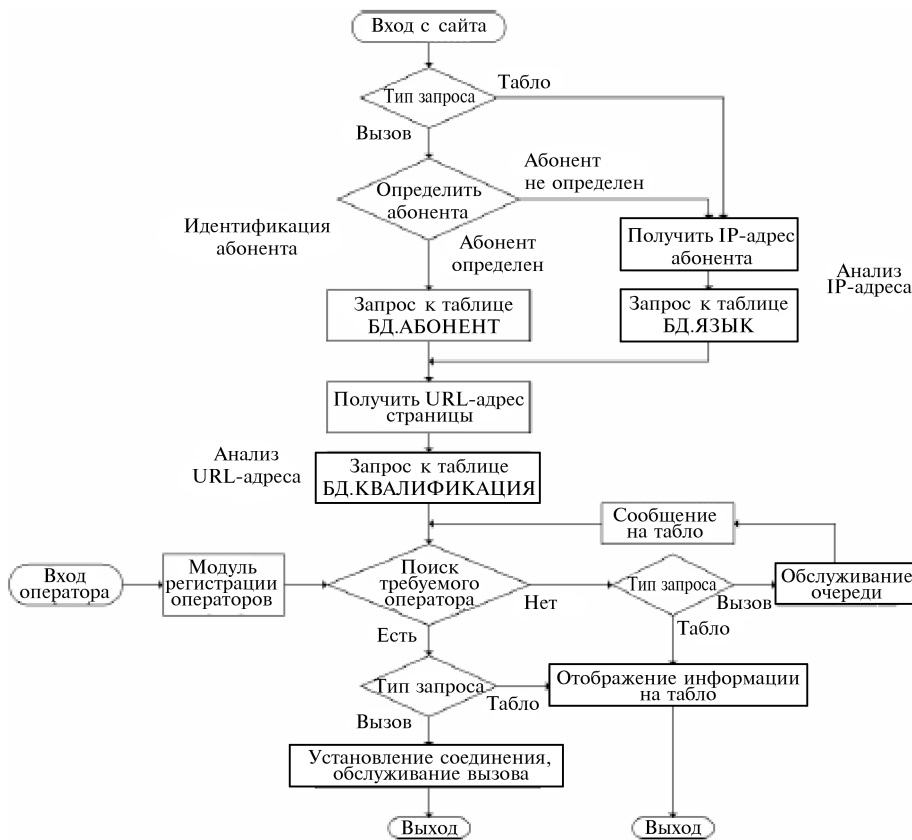


Рис. 2. Алгоритм обработки голосовых вызовов в сети Интернет

На входе в алгоритм со страницы Интернет-сайта на сервер ЦОВ поступает запрос, требующий обновить/отобразить информацию на табло или обработать голосовой вызов абонента. Если абонент инициирует голосовой вызов, то его идентифицируют по наличию на компьютере файла **cookie**, содержащего идентификационный код. Если файл найден и имеет корректный формат, из него получают идентификационный код абонента, по которому, через запрос к таблице "БД. АБОНЕНТ" определяют язык обслуживания абонента и дополнительную информацию. Дополнительная информация предоставляет оператору персональные данные абонента и позволяет не вводить их в базу данных при очередном заказе.

В случае если файл **cookie** отсутствует или имеет некорректный формат, абонента считают не определенным и его вызов поступает в ЦОВ впервые. Файл **cookie** создают на компьютере абонента при делегировании или регистрации идентификационного кода. Если абонент не определен, запрашивают его IP-адрес, преобразуют полученное значение из строкового формата к числовому представлению и по таблице "БД. ЯЗЫК" определяют языковую и территориальную принадлежность абонента.

При анализе URL-адреса страницы, с которой поступил запрос (как в случае вызова, так и для отображения на табло), по таблице "БД. КВАЛИ-

ФИКАЦИЯ" формируют список операторов, которые могут обслуживать требуемую страницу Интернет-сайта на языке абонента.

После регистрации оператора в системе обслуживания вызовов сервер ЦОВ формирует динамическую таблицу состояния операторов. В простейшем случае в разные моменты времени оператор может находиться в состоянии "свободен", "обслуживает вызов", "отсутствует на месте". Сервер ЦОВ формирует список операторов со статусом "свободен" (а также подсчитывает их число, определяет число операторов со статусом "обслуживает вызов", значения которых отображаются на табло) и удовлетворяющих поступившим с Интернет-сайта требованиям. Далее проверяют тип поступившего запроса и при необходимости предоставления информации для табло результат отображают на Интернет-странице, затем выходят из алгоритма.

В случае наличия свободных операторов на табло отображают их число и общее число операторов, которые обслуживают данную страницу на языке абонента. При отсутствии свободных операторов на табло отображают размер очереди и предлагают форму заказа обратного вызова абонента оператором. При постановке вызова в очередь информацию на табло выводят в текстовом или графическом виде, а голосовое соединение не устанавливают.

Когда оператор, требуемый для обслуживания вызова, свободен, голос абонента оцифровывают, сжимают посредством определенного алгоритма и в виде IP-пакетов передают через сеть Интернет в адрес ЦОВ, в ЦОВ IP-пакеты преобразуют обратно в голосовой сигнал, устанавливают голосовое соединение между абонентом и оператором.

Обслуживание очереди вызовов

В отчете сравнительной аналитики *Merchants Global Contact Center Benchmarking Report 2005* [1] исследовательская группа *Dimension Data* представила ряд фактов о роли передовых технологий в практической деятельности ЦОВ по всему миру. Один из выводов данного отчета заслуживает особого внимания — выяснилось, что, хотя сегодня абоненты обращаются в ЦОВ чаще, чем прежде, доля

несостоявшихся вызовов растет третий год подряд. Именно нетерпеливость абонентов стала причиной рекордных 13 % вызовов, прерванных прежде, чем оператор успел на них ответить. Аналитики отметили, что ситуация еще хуже в телекоммуникационной индустрии, где несостоявшимся оказывается каждый пятый телефонный разговор.

Повлиять на качество обслуживания нетерпеливых абонентов позволяет предлагаемый метод обработки вызовов в очереди, заключающийся в визуализации состояния обслуживания (отображение на табло размера очереди, числа операторов и т. д.) без необходимости установления голосового соединения. Принцип работы метода поясняет рис. 3, на котором приведена схема работы очереди.

Анализ интенсивности поступления вызовов и закона распределения времени "нетерпеливости" абонентов в ЦОВ при наличии визуализации состояния обслуживания является наиболее сложным направлением в исследованиях. Для изучения таких параметров целесообразно применить распределение Вейбулла [2], названное в честь шведского инженера, ученого и математика *Ernst Hjalmar Waloddi Weibull* (1887—1979). Применение исследований Вейбулла в части анализа поведения абонентов в ЦОВ описаны в работе [3]. Также для анализа распределения времени терпеливости абонентов возможно использование моделей Эрланг-А и Эрланг-В с отказами [4, 5].

Допустим, что X — случайная величина, характеризующая время ожидания абонента в очереди на обслуживание к оператору ЦОВ. Тогда интенсивность отказа абонентов (нетерпеливость) выражается следующим образом:

$$\lambda_q(x) = \frac{f(x)}{1 - F(x)}, \quad (1)$$

где $F(x)$ и $f(x)$ — соответствующие функции распределения и плотности случайной величины X .

Как правило, отказы от ожидания обслуживания в очереди возникают по типовым причинам и имеют следующие характерные интервалы.

Интервал отказов при вхождении в очередь. Абонент находится в очереди очень непродолжительное время, а затем разрывает соединение. Причиной такого поведения является нежелание находиться в очереди или ошибочно набранный номер. Функция $\lambda_q(x)$, характеризующая такие отказы, имеет сравнительно высокие значения и

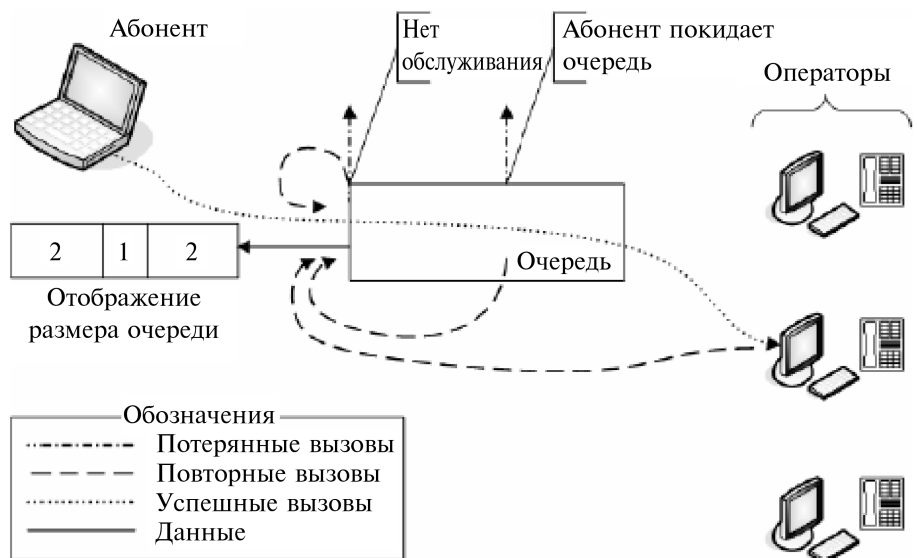


Рис. 3. Обслуживание очереди с визуализацией состояния обработки вызовов

явную тенденцию к убыванию (чаще всего она убывает монотонно).

Интервал отказов при нормальной обработке вызова в очереди. При таком обслуживании прогноз времени ожидания ответа оператора, который сообщают абоненту в очереди, оказывается приемлемым. В системе остаются только те абоненты, которым необходимо связаться с оператором, и они готовы затратить определенное время на ожидание. Функция $\lambda_q(x)$ на данном интервале характеризуется стабильностью и сравнительно низкой интенсивностью отказов.

Интервал окончания терпеливости абонентов. Отказ от обслуживания происходит, как правило, при получении повторных сообщений с неудовлетворительным прогнозом времени ожидания или по окончании периода терпимости абонентов. Отказы от обслуживания в этот период связаны с истинной причиной вызова и нервно-психической устойчивостью абонента.

Каждому такому интервалу соответствует определенный вид функции интенсивности отказов $\lambda_q(x)$. Рассмотрим класс степенных зависимостей

$$\lambda_q(x) = \lambda_0 \beta x^{\beta - 1}, \quad (2)$$

где $\lambda_0 > 0$ и $\beta > 0$ — некоторые числовые параметры, характеризующие интенсивность отказов в различные интервалы времени.

Соотношение (1) при заданной интенсивности потери вызовов $\lambda_q(x)$ является дифференциальным уравнением относительно функции $F(x)$. Из теории дифференциальных уравнений следует, что

$$F(x) = 1 - \exp\left\{-\int_0^x \lambda_q(t) dt\right\}. \quad (3)$$

Подставив (2) в (3), получим, что

$$F(x) = \begin{cases} 1 - e^{(-\lambda_0 x^\beta)}, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (4)$$

Распределение, задаваемое формулой (4), является распределением Вейбулла, поскольку, обозначив $\lambda_0 x^\beta = \left(\frac{x}{\alpha}\right)^\beta$, получим

$$F(x) = \begin{cases} 1 - e^{-\left(\frac{x}{\alpha}\right)^\beta}, & x \geq 0, \\ 0, & x < 0. \end{cases} \quad (5)$$

Величина $\alpha = 1/\beta\sqrt[\beta]{\lambda_0}$, введенная в формуле (5), является параметром масштаба распределения, а параметр β является коэффициентом формы распределения.

Влияние масштабного множителя α аналогично влиянию стандартного отклонения нормального распределения, т. е. с увеличением α происходит выравнивание кривой распределения. Коэффициент β влияет на асимметрию распределения. Значения $0 < \beta < 1$ соответствуют интервалу вхождения вызовов в очередь, при $\beta = 1$ происходит рабочее обслуживание вызовов в очереди, а при $\beta > 1$ наступает окончание времени терпимости абонентов.

Плотность распределения Вейбулла определяется формулой

$$f(x) = \begin{cases} \frac{\beta}{\alpha} \left(\frac{x}{\alpha}\right)^{\beta-1} e^{-\left(\frac{x}{\alpha}\right)^\beta}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

Зависимость, отражающая число вызовов, остающихся на обслуживании в очереди ЦОВ через время x , определяется по формуле

$$R(x) = 1 - F(x) = e^{-\left(\frac{x}{\alpha}\right)^\beta}.$$

Математическое ожидание (среднее значение терпимости абонентов) при распределении вызовов в соответствии с законом Вейбулла можно вычислить по формуле

$$E[X] = \alpha \Gamma\left(1 + \frac{1}{\beta}\right),$$

где $\Gamma(x) = \int_0^\infty z^{x-1} e^{-z} dz$ — гамма-функция, значение которой табулировано.

Для оценки значений коэффициентов α и β распределения необходимо провести следующую последовательность действий:

- вычислить выборочное среднее арифметическое значение \bar{x} и выборочное среднее квадратичное отклонение $\bar{\sigma}$ по формулам

$$\bar{x} = \frac{1}{m} \sum_{i=1}^m x_i, \text{ и } \bar{\sigma} = \sqrt{\frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^2};$$

- далее вычислить коэффициент асимметрии ρ по формуле

$$\rho = \frac{\frac{m}{(m-1)(m-2)} \sum_{i=1}^m (x_i - \bar{x})^3}{\left[\frac{1}{m-1} \sum_{i=1}^m (x_i - \bar{x})^2\right]^{3/2}};$$

- по полученным значениям ρ из таблицы находят оценку параметра β и значения коэффициентов g и K ;
- оценочное значение параметра α определяют по формуле $\alpha = \bar{\sigma}/g$.

Проведение статистического анализа поведения абонентов в очереди требует постоянства условий обработки анализируемых вызовов во времени. Изменение технологических параметров обслуживания в процессе эксплуатации ЦОВ (увеличение времени обработки вызова, изменение тематики диалога абонента с оператором, условий распределения вызовов и т. п.) может существенно изменить динамику терпимости абонентов в очереди. Также необходим тщательный анализ отказов от обслуживания при

Значения g , K и β для заданных значений ρ

ρ	K	g	β
190,1	120	1901	0,20
6,615	2,00	4,472	0,50
2,00	1,00	1,00	1,00
1,734	0,965	0,878	1,10
1,521	0,94	0,787	1,20
1,346	0,923	0,716	1,30
1,198	0,911	0,66	1,40
1,072	0,903	0,613	1,50
0,962	0,897	0,574	1,60
0,865	0,892	0,54	1,70
0,779	0,889	0,511	1,80
0,701	0,888	0,486	1,90
0,631	0,886	0,463	2,00
0,567	0,886	0,443	2,10
0,509	0,886	0,425	2,20
0,455	0,886	0,408	2,30
0,405	0,886	0,393	2,40
0,358	0,887	0,38	2,50
0,315	0,888	0,367	2,60
0,275	0,889	0,355	2,70
0,237	0,89	0,344	2,80
0,202	0,891	0,333	2,90
0,168	0,893	0,325	3,00
-0,087	0,906	0,254	4,00
-0,254	0,918	0,21	5,00
-0,373	0,928	0,18	6,00
-0,463	0,935	0,157	7,00
-0,534	0,942	0,14	8,00
-0,591	0,947	0,126	9,00
-0,638	0,951	0,114	10,00

вхождении абонентов в очередь. Причины таких отказов могут отличаться для различных категорий ЦОВ. Раннее отключение абонента может происходить как по причине нарушения технологии установления голосового соединения, так и вследствие особенностей поведения различных категорий обслуживаемых абонентов.

В 1876 году Александру Беллу удалось передать речь из одной точки в другую посредством электрического сигнала, именно это событие связывают с изобретением телефона. Тогда мало кто мог предположить, что через 50 лет такой вид связи будет активно использоваться не только в личных целях, но и в бизнесе, а еще через 40 лет появятся системы автоматического распределения телефонных вызовов между операторами ЦОВ. Поскольку развитие технологий и общества происходит по спирали, то будет вполне логичным предположить подобные пути эволюции голосовой связи в сетях передачи данных с коммутацией пакетов.

Удешевление голосовых сервисов в сетях передачи данных и повышение качества связи на фоне увеличения общих масштабов обслуживания абонентов дают основание ожидать увеличения доли вызовов с коммутацией пакетов, обрабатываемых в центрах обработки вызовов сети Интернет. Предложенные в статье методы исследованы в ходе экспериментов, которые подтвердили предпо-

ложения о повышении эффективности обслуживания. Практическая ценность описанных методов дает возможность их применения для предоставления как услуг аутсорсинга обработки вызовов в сети Интернет, так и сервиса аренды Интернет-приложений ЦОВ.

В настоящее время продолжается совершенствование функциональности ЦОВ и, соответственно, поиск дополнительных возможностей предоставления клиентам современных сервисов обслуживания. При этом неизбежно возникают новые и усложняются существующие задачи ЦОВ, решение которых требует разработки соответствующих подходов к организации работы с клиентами и абонентами. Дальнейшее развитие обслуживания абонентов возможно в направлении децентрализации операторских и аппаратных ресурсов за счет доступности сети Интернет в большинстве стран и регионов.

Список литературы

1. **Доусон К.** Мировая индустрия call-центров в цифрах и фактах // Сети и системы связи. 2005. № 14.
2. **Weibull W.** A statistical distribution function of wide applicability // J. Appl. Mech.-Trans. ASME 18 (3). 1951. P. 293—297.
3. **Kort B. W.** Models and methods for evaluating customer acceptance of telephone connections // Proc. of IEEE Globecom'83. 1983. P. 706—714.
4. **Mandelbaum A., Zeltyn S.** The Palm/Erlang-A Queue, with Applications to Call Centers // Faculty of Industrial Engineering & Management, Technion, 2005.
5. **Росляков А. В., Ваняшин С. В.** Математические модели центров обслуживания вызовов. М.: ИРИАС. 2006. 336 с.

УДК 004.52

Шахов В. Г., канд. техн. наук, проф.,
Нопин С. В.,
Омский государственный университет
путей сообщения

Использование IP-телефонии в защищенном режиме

Рассмотрены варианты реализации речевой IP-связи в контексте организации ее конфиденциальности. Приведены варианты соединений с их критическим анализом. Дан анализ протоколов взаимодействия с выбором эффективных решений.

Тенденции развития современной цивилизации имеют значительный дифферент в сторону информатизации и оказания информационных услуг. В этой области концентрируются большие

капиталовложения, а сама отрасль развивается очень бурно. Среди множества приложений в области информационного сервиса значительное место занимает обслуживание речевых взаимодействий пользователей. Сравнительно новым направлением является IP-телефония — очень серьезный конкурент в среде обеспечения речевых взаимодействий (классическая телефонная связь, в том числе цифровая, радиосвязь, включая сотовую, мультимедийные приложения информационных потоков) [12]. Преимущества IP-телефонии проявляются при налаженной структуре взаимодействий, так как она накладывается на уже существующее "железо" поверх созданных корпоративных сетей, систем удаленного пользования, систем связи и коммуникаций.

На современном этапе система IP-телефонии основана на своих стандартах сжатия речи, форматах и протоколах передачи, драйверах согласования пакетов с провайдерами. От последних тре-

буются сравнительно небольшие дополнительные затраты на оборудование, которые быстро окупаются при наличии клиентов [11].

Возникает проблема защищенной речевой связи, причем эта услуга должна оплачиваться клиентами. Провайдер, оказывающий подобные услуги, должен гарантировать следующие условия [7]:

- невозможность восстановления передаваемой речевой информации в течение гарантированного срока со стороны сторонних клиентов или при перехвате сообщений;
- невозможность фальсификации речевой информации после ее перехвата;
- доступность сравнительно легкого восстановления информации специальными государственными органами при наличии у них соответствующих полномочий.

Последняя фраза может показаться спорной, но в противном случае, ни одно государство не сертифицирует такой способ информационного обмена. Понятно, что сама идея защищенности серьезно компрометируется, но надежно закрытая речевая связь может гарантироваться государством.

Рассматриваемая проблема имеет свою терминологию, учитывающую степени защиты передаваемой информации [14]. **Секретная** связь обеспечивает информационные взаимодействия, обеспечивающие закрытые каналы в интересах государства. **Защищенная** (или конфиденциальная) связь поддерживает режимы взаимодействия в пределах полномочий соответствующего охранного документа. Полномочия и срок действия документа оплачивает лицо, купившее соответствующую услугу. **Маскированная** связь оплачивается конкретным пользователем без наличия соответствующих охранных документов, гарантирующих качество услуг.

В предлагаемых материалах обсуждаются технические аспекты создания, оценки качества и тестирования систем конфиденциальной связи в IP-режиме, а также способы тестирования и проверки качества конфиденциальной IP-связи.

Технология передачи речевой информации по сетям с маршрутизацией пакетов IP привлекает, в первую очередь, своей универсальностью — речь может быть преобразована в поток IP-пакетов в любой точке сетевой инфраструктуры: на магистрали сети оператора, на границе территориально распределенной сети, в корпоративной сети и даже непосредственно в терминале конечного пользователя [4]. Несмотря на универсальность протокола IP, внедрение систем IP-телефонии сдерживается тем, что многие операторы считают их недостаточно надежными, плохо управляемыми и не очень эффективными. Но грамотно спроектированная сетевая инфраструктура с эффективными механизмами обеспечения качества обслужи-

вания делает эти недостатки малосущественными. В расчете на порт стоимость систем IP-телефонии находится на уровне стоимости систем *Frame Relay* и заведомо ниже стоимости оборудования ATM [2].

Архитектура технологии *Voice over IP* может быть упрощенно представлена в виде двух плоскостей. Нижняя плоскость — это базовая сеть с маршрутизацией пакетов IP, верхняя — это открытая архитектура управления обслуживанием вызовов (запросов связи).

Нижняя плоскость представляет собой комбинацию известных протоколов Интернет: RTP (*Real Time Transport Protocol*), который функционирует поверх протокола UDP (*User Datagram Protocol*), расположенного, в свою очередь, в стеке протоколов TCP/IP над протоколом IP. Таким образом, иерархия RTP/UDP/IP представляет собой своего рода транспортный механизм для речевого трафика.

Верхняя плоскость управления обслуживанием вызова предусматривает принятие решений о том, куда вызов должен быть направлен, и каким образом должно быть установлено соединение между абонентами. Инструмент такого управления — телефонные системы сигнализации, начиная с систем, поддерживаемых декадно-шаговыми АТС и предусматривающих объединение функций маршрутизации и функций создания коммутируемого разговорного канала в одних и тех же декадно-шаговых искателях.

В сетях с коммутацией пакетов ситуация более сложная. Сеть с маршрутизацией пакетов IP принципиально поддерживает одновременно целый ряд разнообразных протоколов маршрутизации. Такими протоколами на сегодня являются: RIP (*Routing Information Protocol*), IGRP (*Interior Gateway Routing Protocol*), EIGRP (*Enhanced Interior Gateway Routing Protocol*), IS-IS (*Intermediate System-to-Intermediate System*), OSPF (*Open Shortest Path First*), BGP (*Border Gateway Protocol*) и др. Точно так же и для IP-телефонии разработан целый ряд протоколов [7].

Первый в истории и наиболее распространенный подход к построению сетей IP-телефонии на стандартизированной основе предложен Международным союзом электросвязи ИТУ в рекомендации H.323 [6]. Рекомендация предусматривает довольно сложный набор протоколов, который предназначен не просто для передачи речевой информации по IP-сетям с коммутацией пакетов. Его цель — обеспечение работы мультимедийных приложений в сетях с негарантированным качеством обслуживания. Речевой трафик — это только одно из приложений H.323, наряду с видеоинформацией и данными.

Вариант построения сетей IP-телефонии, предложенный Международным союзом электро-

связи в рекомендации Н.323, хорошо подходит тем операторам местных телефонных сетей, которые заинтересованы в использовании сети с коммутацией пакетов (IP-сети) для предоставления услуг междугородной и международной связи.

Другой протокол плоскости управления обслуживанием вызова — SIP (*Session Initiation Protocol*) — ориентирован на то, чтобы сделать оконечные устройства и шлюзы более интеллектуальными и поддерживать дополнительные услуги для пользователей. Подход SIP к построению сетей IP-телефонии намного проще в реализации, чем Н.323, но меньше подходит для организации взаимодействия с телефонными сетями. В основном это связано с тем, что протокол сигнализации SIP, базирующийся на протоколе H.323, плохо согласуется с системами сигнализации, используемыми в телефонных сетях общего пользования (ТфСОП).

Третий подход к построению сетей IP-телефонии основан на использовании протокола MGCP. Сетевая архитектура данного протокола содержит следующие основные функциональные блоки:

- шлюз Media Gateway, который выполняет функции преобразования речевой информации, поступающей со стороны ТфСОП с постоянной скоростью передачи, в вид, пригодный для передачи по сетям с маршрутизацией пакетов IP (кодирование и упаковку речевой информации в пакеты RTP/UDP/IP, а также обратное преобразование);
- контроллер шлюзов Call Agent, которой выполняет функции управления шлюзами;
- шлюз сигнализации Signaling Gateway (SG), который обеспечивает доставку сигнальной информации, поступающей со стороны ТфСОП, к контроллеру шлюзов и перенос сигнальной информации в обратном направлении.

Особенностью протокола является то, что весь интеллект функционально распределенного шлюза сосредоточен в контроллере, функции которого могут быть распределены между несколькими компьютерными платформами.

При построении систем IP-телефонии наиболее часто используют три варианта систем IP-телефонии [10]: компьютер — компьютер; компьютер — телефон; телефон — телефон.

Сценарий "компьютер—компьютер" (рис. 1, см. четвертую сторону обложки) реализуется на базе стандартных компьютеров, оснащенных средствами мультимедиа и подключенных к вычислительной сети. В этом сценарии аналоговые речевые сигналы от микрофона абонента А преобразуются в цифровую форму с помощью аналого-цифрового преобразователя (АЦП), обычно при 8000 отсчетов/с, 8 бит/отсчет, в итоге — 64 Кбит/с. Отсчеты речевых данных в цифровой форме могут затем сжиматься кодирующим устройством для

сокращения нужной для их передачи полосы. Выходные данные после сжатия формируются в пакеты, к которым добавляются заголовки протоколов, после чего пакеты передаются через IP-сеть в систему IP-телефонии, обслуживающую абонента Б. Когда пакеты принимаются системой абонента Б, заголовки протокола удаляются, а сжатые речевые данные поступают в устройство, развертывающее их в первоначальную форму, после чего речевые данные снова преобразуются в аналоговую форму с помощью цифроаналогового преобразователя (ЦАП) и попадают в телефон абонента Б. Для обычного соединения между двумя абонентами системы IP-телефонии на каждом конце одновременно реализуют как функции передачи, так и функции приема.

Данный сценарий ориентирован на пользователя, которому сеть нужна для передачи данных, а программное обеспечение IP-телефонии требуется лишь иногда для разговоров. Эффективное использование телефонной связи по сценарию "компьютер — компьютер" обычно связано с повышением продуктивности работы крупных компаний, например, при организации виртуальной презентации в корпоративной сети с возможностью не только видеть документы на Web-сервере, но и обсуждать их содержание с помощью IP-телефона. При этом между двумя IP-сетями могут использоваться элементы ТфСОП. Наиболее распространенным программным обеспечением для этих целей является пакет Microsoft NetMeeting, доступный для бесплатной загрузки с узла Microsoft.

Следующий сценарий — "телефон — компьютер" (рис. 2, см. четвертую сторону обложки) находит применение в разного рода справочно-информационных службах сети Интернет, в службах сбыта товаров или службах технической поддержки. Пользователь, подключившийся к серверу WWW какой-либо компании, имеет возможность обратиться к оператору справочной службы. Компании будут использовать данную технологию для наращивания своего присутствия во всемирной паутине. Это вполне соответствует стилю жизни современных потребителей, связанному с потребностью в дополнительных удобствах и экономии времени.

Популярными программными продуктами для этого варианта сценария IP-телефонии "компьютер — телефон" являются Net2Phone и Skype, организующие вызовы к обычным абонентским телефонным аппаратам в любой точке мира.

Сценарий "телефон — телефон" (рис. 3, см. четвертую сторону обложки) в значительной степени отличается от остальных сценариев IP-телефонии своей социальной значимостью, поскольку целью его применения является предоставление обычным абонентам ТфСОП альтернативной

возможности междугородной и международной телефонной связи. В этом режиме современная технология IP-телефонии предоставляет виртуальную телефонную линию через IP-сети.

Как правило, обслуживание вызовов по такому сценарию IP-телефонии выглядит следующим образом. Поставщик услуг IP-телефонии подключает свой шлюз к коммутационному узлу или станции ТфСОП, а по сети Интернет или по выделенному каналу соединяется с аналогичным шлюзом, находящимся в другом городе или другой стране.

Типичная услуга IP-телефонии по сценарию "телефон — телефон" использует стандартный телефон в качестве интерфейса пользователя, а вместо междугородного компонента ТфСОП использует либо частную IP-сеть/Intranet, либо сеть Интернет. Благодаря маршрутизации телефонного трафика по IP-сети стало возможным обходить сети общего пользования и, соответственно, не платить за междугородную/международную связь операторам этих сетей.

Поставщики услуг IP-телефонии предоставляют услуги "телефон—телефон" путем установки шлюзов IP-телефонии на входе и выходе IP-сетей. Абоненты подключаются к шлюзу поставщика через ТфСОП, набирая специальный номер доступа. Абонент получает доступ к шлюзу, используя персональный идентификационный номер (PIN). После этого шлюз просит ввести телефонный номер вызываемого абонента, анализирует этот номер и определяет, какой шлюз имеет лучший доступ к нужному телефону. Как только между входным и выходным шлюзами устанавливается контакт, дальнейшее установление соединения к вызываемому абоненту выполняется выходным шлюзом через его местную телефонную сеть.

Общая задержка речевой информации делится на две основные части — задержка при кодировании и декодировании речи в шлюзах или терминальном оборудовании пользователей и задержка, вносимая самой сетью. Уменьшить общую задержку можно двумя путями: во-первых, спроектировать инфраструктуру сети таким образом, чтобы задержка в ней была минимальной, и, во-вторых, уменьшить время обработки речевой информации терминальным оборудованием пользователей.

Наиболее точно соответствующей результатам измерения качества передачи голоса методом субъективной экспертной оценки (MOS), сегодня считается методика PESQ (Perceptual Evaluation of Speech Quality, ITU-T P.862). Значение PESQ является интегральным показателем качества и включает в себя широкий круг параметров, влияющих на качество, в том числе, деградацию голоса вследствие использования тех или иных

кодеков, ошибок передачи, потери пакетов, значения задержки передачи пакетов и ее вариаций (джиттер), и рассогласования на стыках цифровых и аналоговых элементов сетевой инфраструктуры.

Привлекательность всех алгоритмов сценария "телефон — телефон" для пользователя заключается в значительно более низких, по сравнению с обычной междугородной или международной телефонной связью тарифах, что является следствием применения технологий, обеспечивающих вторичное уплотнение каналов передачи данных. Кроме того, для алгоритмов сценария "компьютер — компьютер" существует возможность организовывать сеанс связи в режиме, защищенном от несанкционированного доступа. Поэтому многие пользователи согласны терпеть снижение качества передачи речи.

Предоставление телефонных услуг через инфраструктуру IP позволяет поставщику услуг IP получать большую по сравнению с традиционными операторами прибыль благодаря тому, что:

- функции предоставления услуг телефонии и передачи данных объединяются в общей инфраструктуре IP; основной объем обслуживаемого трафика приходится на традиционные данные сети Интернет, а транспортировка относительно невысокого объема трафика IP-телефонии может осуществляться с использованием той же инфраструктуры при очень незначительных дополнительных затратах;
- отсутствует необходимость обеспечения качества и объема услуг, требуемых от операторов ТфСОП, что допускает реализацию услуг IP-телефонии на базе более дешевого оборудования.

На уровне передачи данных пакеты, передаваемые по двухточечной линии, могут кодироваться при передаче в линию и декодироваться при приеме. Этот метод, называемый шифрованием в канале связи, легко может быть добавлен к любой сети. Все детали шифрования могут быть известны только уровню передачи данных. Однако такое решение перестает работать в том случае, если пакету нужно преодолеть несколько маршрутизаторов, поскольку при этом пакет придется расшифровывать на каждом маршрутизаторе, что сделает его беззащитным перед атаками внутри маршрутизатора. Кроме того, такой метод не позволит защищать отдельные сеансы, требующие защиты, и при этом не защищать остальные.

На сетевом уровне могут быть установлены брандмауэры, позволяющие отвергать подозрительные пакеты, приходящие извне.

На транспортном уровне можно зашифровать соединения целиком, от одного конца до другого. Максимальную защиту может обеспечить только такое сквозное шифрование.

Наконец, проблемы аутентификации и обеспечения строгого выполнения обязательств могут решаться только на прикладном уровне.

Итак, очевидно, что безопасность в сетях — это вопрос, охватывающий все уровни протоколов.

На всех уровнях, за исключением физического, защита информации базируется на принципах применения специальных целочисленных методов, предоставляющих определенный минимум гарантий защиты. Задача исследования — выбор методов защиты, обеспечивающих должные гарантии пользователей при соблюдении должных норм взаимодействия частных лиц и государственных служб.

Список литературы

1. **Васильева Е. Н.** Анализ и разработка комплексного обеспечения помехоустойчивой передачи информации и защиты от несанкционированного доступа. Автореф. дис. ... канд. техн. наук / Е. Н. Васильева. — СПб., 2001. — 16 с.
2. **Вегешна Ш.** Качество обслуживания в сетях IP / Ш. Вегешна. — М.: Вильямс, 2003. — 368 с.
3. **Гольдштейн Б. С.** IP-телефония / Б. С. Гольдштейн, А. В. Пинчук, А. Л. Суховицкий. — М.: Радио и связь, 2001. — 336 с.
4. **Гольдштейн Б. С.** Call-центры и компьютерная телефония / Б. С. Гольдштейн, В. А. Фрейкман. СПб.: BHV-Санкт-Петербург, 2002. — 370 с.
5. **ГОСТ 16600—72.** Передача речи по трактам радиотелефонной связи. Требования по разборчивости речи и методы артикуляционных измерений. — М.: Издательство стандартов, 1972.

6. **Иванова Т. И.** Протокол H.323 в сетях IP-телефонии / Т. И. Иванова — М.: ТСС, 2000, № 2. С. 24—35.

7. **Кучников Т. В.** Интернет-телефония / Т. В. Кучников — М.: Альянс-пресс, 2004. — 128 с.

8. **Нопин С. В.** Моделирование защиты речевой информации с помощью персонального компьютера / С. В. Нопин, В. Г. Шахов, Д. А. Бесов // Научная сессия МИФИ-2006. XIII Всероссийская научная конференция "Проблемы информационной безопасности в системе высшей школы". Сборник научных трудов. М.: МИФИ, 2006. С. 88—89.

9. **Нопин С. В.** Преобразование речевой информации с помощью Audio Compression Manager (ACM) в системах IP-телефонии / С. В. Нопин, В. Г. Шахов // Актуальные проблемы развития железнодорожного транспорта, материалы II Международной научно-практической конференции 7—8 декабря 2005 года. Самара: СамГАПС, 2006. С. 174—176.

10. **Нопин С. В.** Возможности использования встроенных звуковых кодеков операционной системы (ОС) Windows в системах IP-телефонии / С. В. Нопин, В. Г. Шахов // Омский научный вестник. 2006, № 1 (34). С. 155—157.

11. **Остерлох Х.** Маршрутизация в IP-сетях. Принципы, протоколы, настройка. / Х. Остерлох — Киев: ДиаСофтЮП, 2002. — 512 с.

12. **Росляков А. В.** IP-телефония / А. В. Росляков, М. Ю. Самсонов, И. В. Шibaева. — М.: Эко-Трендз, 2003. — 252 с.

13. **Шелухин О. И.** Цифровая обработка и передача речи: Кодеры CELP, LD-CELP, ACELP, MBE, MBEV. IP-телефония, ATM, FrameRelay, Internet. Кодеры мобил. систем радиосвязи GSM, TETRA, APCO-25, INMARSAT-M. Системы спутниковой связи VSAT: DAMA, PAMA. Системы TDMA, FDMA, ALOXA, S-ALOXA / О. И. Шелухин, Н. Ф. Лукьянцев. — М.: Радио и связь, 2000. — 454 с.

14. **Шеннон К. Э.** Теория связи в секретных системах. / К. Э. Шеннон // В кн.: Шеннон К. Э. Работы по теории информации и кибернетике. М.: ИЛ, 1963. С. 333—402.



WEB-ТЕХНОЛОГИИ

УДК 004.738.52

С. В. Свечников, ФГУ ГНИИИ ИТТ "Информика"

Высокорелевантный поиск и автоматическая категоризация ресурсов сети Интернет

Предложен подход для реализации высококорелевантного поиска и автоматической категоризации интернет-ресурсов. Рассматривается механизм индексации интернет-ресурсов, т. е. преобразования их в единый формат, посредством выделения терминов и присвоения им весовых коэффициентов, что позволяет достаточно быстро и эффективно оценить контент интернет-ресурса и выделить главное. Рассматривается задача, связанная с автоматической категоризацией интернет-ресурсов, обучением системы и отнесением текстовой информации к заранее определенной категории за счет использования степени соответствия между категорией и интернет-ресурсом.

За последние несколько лет активное развитие информационно-коммуникационных технологий привело к тому, что объем информационных ресурсов значительно вырос. Этот всевозрастающий объем информации, а также ее различные виды представлений (текстовая, графическая, аудио-,

видеоинформация) приводят к проблемам, связанным с неудобством при работе с ней, а также с бесконтрольным доступом к сети Интернет.

В сегодняшнем деловом мире использование сети Интернет является неотъемлемым процессом, так же как растет число пользователей Ин-

тернета, так растет и количество доступной информации в нем. Наряду с преимуществами сеть Интернет является и самым большим источником опасностей — почти каждый день появляются новые вредоносные материалы, такие как спам, агрессивный контент и шпионские программы. Российский сегмент сети — один из самых быстроразвивающихся, число пользователей сети Интернет в России по различным данным около 25 млн чел., из них 2 млн детей [6].

Обеспечение учебных заведений и публичных библиотек доступом к сети Интернет увеличивает число учащихся, пользующихся различными сервисами и информационными источниками, предоставляемыми глобальной сетью. Такой бесконтрольный доступ к сети Интернет может привести к серьезным угрозам для детей и учащихся. Также Интернет бесконтрольно используется в личных целях работниками умственного труда, имеющими доступ к глобальной сети, что снижает эффективность их работы и производительность корпоративной сети [1]. При этом методы прямого регулирования (цензуры) неэффективны, встречают протест пользователей Интернета и юридически несостоятельны, поскольку противоречат естественным правам граждан на свободу воли, высказываний и волеизъявления.

В связи с этим решение данной проблемы надо искать не в цензуре, а в предоставлении инструмента для защиты от нежелательного контента, который пользователи могут использовать по своей воле и по своему усмотрению [2]. Реализация такого инструмента тесно связана с тематической категоризацией интернет-ресурсов, которая имеет много важных и востребованных в современном мире применений. Одним из них является механизм фильтрации полезной информации от агрессивного контента. При этом все интернет-ресурсы разбиваются на категории и после соответствующего обучения становится возможной автоматическая фильтрация.

Основные представленные на российском рынке программные продукты в области тематической категоризации интернет-ресурсов и их производители представлены в таблице.

Другие системы или не поддерживают фильтрацию русскоязычных интернет-ресурсов или являются непригодными для корпоративной эксплуатации.

Производитель	Страна	Программный продукт
Secure computing	США	Sentian
Surfcontrol	США	Surfcontrol web-filter
Websense	США	Websense Enterprise
Cobion	Германия	Proventia Web Filter

Перечисленные решения представляют собой программы, которые устанавливаются в локальной сети организации и работают на принципе анализа и тематической категоризации интернет-ресурсов.

Эти программные продукты позволяют:

- повысить эффективность работы сотрудников;
- оптимизировать пропускную способность сети;
- усилить безопасность сети;
- предотвратить появление неэтичных материалов;
- снизить расходы за доступ в сеть Интернет.

К достоинствам представленных программных продуктов можно отнести:

- обеспечение защиты от основных актуальных угроз безопасности;
- высокий уровень настраиваемости;
- поддержку большой и постоянно обновляемой базы данных URL;
- простоту использования.

Все вышеуказанные системы прекрасно фильтруют, в первую очередь, англоязычный контент. При работе с русскоязычным контентом эти продукты демонстрируют:

- некорректную работу с контентом по причине отсутствия специальных инструментов работы с информацией на русском языке;
- бедность базы данных русскоязычных ресурсов по причине, указанной выше, а также, потому что Рунет составляет 5 % от мировой сети Интернет и, вероятно, не является зоной первоочередных интересов иностранных компаний;
- неучет национальных особенностей при категоризации русскоязычных ресурсов, потому что иностранные компании не всегда адекватно учитывают специфику и политические реалии;
- систематическую погрешность категорирования сайтов, связанную, как правило, с полностью автоматическим определением категорий русскоязычных сайтов;
- низкую оперативность обновления;
- слабое сопровождение программного продукта;
- отсутствие возможности объединения данных от разных организаций с целью консолидации отчетности, потому что существующие системы рассчитаны на локальное использование в коммерческих, по большей части, организациях.

В связи с представленными недостатками существует необходимость создания системы для высокорелевантного поиска и тематической категоризации, адаптируемой для русскоязычных интернет-ресурсов.

Реализация системы высокорелевантного поиска и тематической категоризации интернет-ресурсов предполагает решение следующих задач:

- индексация интернет-ресурсов (преобразование интернет-ресурсов к единому формату);

- автоматическая категоризация интернет-ресурсов, обучение системы и отнесение текстовой информации к заранее определенной категории;
- оценка качества категоризации с использованием метрик из информационного поиска.

Представленные подзадачи связаны, в первую очередь, с анализом текстовой информации веб-страницы, т. е. с ее содержанием (контентом).

Пусть дано множество интернет-ресурсов D , разделенное на два непересекающихся подмножества T_r и T_s , называемых обучающей и тестовой выборкой. На основании обучающей выборки строится классификатор категорий, а на тестовой выборке проверяется качество категоризации. Пусть также дано соответствие между интернет-ресурсами и некоторой категорией c в виде $\Phi: D \rightarrow \{0, 1\}$, устанавливающее для каждого интернет-ресурса значение 1 в случае принадлежности интернет-ресурса категории, и 0 — в противном случае [2, 3, 4, 7].

Необходимо построить, используя только информацию из обучающей выборки T_r , функцию $\Phi': D \rightarrow \{0, 1\}$, аппроксимирующую Φ , чтобы число ошибок E на тестовой выборке T_s было наименьшим:

$$E = \sum_{T_s} |\Phi - \Phi'| \rightarrow \min. \quad (1)$$

Пусть T — множество терминов, каким-либо образом выделенное из интернет-ресурсов категории c . Тогда интернет-ресурс можно представить в виде терминологического вектора:

$$d_j = (w_{1j}, \dots, w_{|T|j})^T, \quad (2)$$

где w_{ij} — вес термина t_i в интернет-ресурсе d_j .

Описания каждой из категорий представим в виде векторов той же размерности, что и векторы интернет-ресурсов:

$$c = (c_1, \dots, c_{|T|})^T, \quad (3)$$

где c_i — вес термина t_i в описании категории c .

При таком подходе существуют два ограничения:

- нет дополнительной информации о категориях, к которым прикрепляются интернет-ресурсы;
- нет никакой внешней информации о интернет-ресурсе, кроме той информации, которая содержится в нем.

Каждый интернет-ресурс — это вектор, где номера терминов (слов) — его координаты, а веса терминов — значения координат, размерность вектора — это число терминов, встречающихся в интернет-ресурсе. Так как учитываются все термины, векторы получаются большого размера, что затрудняет процесс индексации, поэтому не-

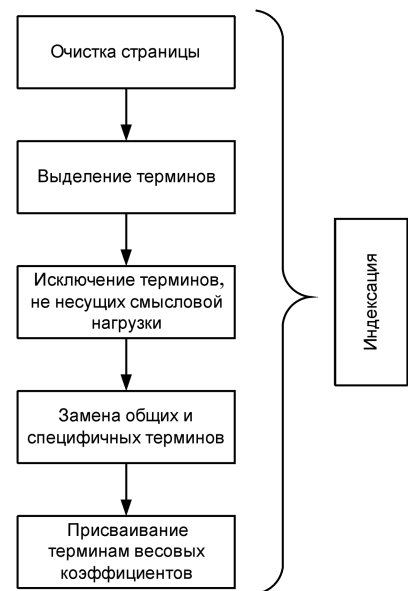


Рис. 1. Индексация интернет-ресурсов

обходимо уменьшить размерность вектора. Для уменьшения размерности вектора в качестве терминов используем не слова, а устойчивые словосочетания, не учитываем редкие слова, которые не несут полезной информации, не рассматриваем часто встречающиеся слова.

Процесс индексации представим следующим образом (рис. 1):

- очистка страницы;
- выделение терминов;
- исключение терминов, не несущих смысловой нагрузки;
- замена общих и специфичных терминов;
- присваивание терминам весовых коэффициентов.

Сначала проводится очистка страницы интернет-ресурса, т. е. удаляется навигационная часть, теги html, скрипты, стоп-слова — частотные слова языка, не несущие смысловой нагрузки (предлоги, союзы, частицы, местоимения, некоторые глаголы), за счет этого уменьшается объем поисковой базы и повышается производительность поиска [9—11]. После этого в тексте с помощью функции анализа контента интернет-ресурса выделяются термины — логические выражения, состоящие из слов и словосочетаний, связанные операторами AND, OR, NOT. Для исключения терминов, не несущих смысловой нагрузки, используется пометка "исключение", которая показывает, что термин не относится к теме. Оставшиеся термины могут также иметь недостатки: существуют термины, которые слишком специфичны или, наоборот, — значение, которых слишком общие, поэтому их необходимо заменить более подходящими. Это увеличивает полноту индексирования. Для замены специфичных терминов используется тезаурус RCO, который представляет собой словарь общей лексики с семантическими

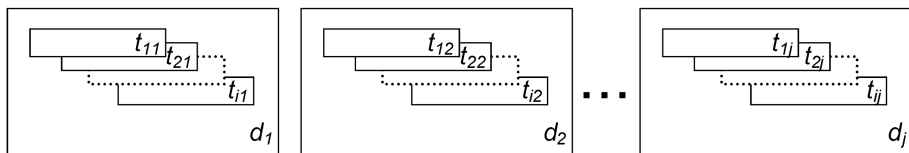


Рис. 2. Интернет-ресурсы, содержащие некоторые термины

отношениями между словами [8]. Использование тезауруса повышает качество анализа текста и полноту поиска информации, позволяя расширять запрос синонимичными, более общими и более частными понятиями. Общие термины заменяются сочетаниями терминов или несколькими связанными терминами, имеющими более определенное значение.

Заключительным этапом является присвоение терминам весовых коэффициентов. Исходное представление интернет-ресурса выглядит следующим образом: интернет-ресурс = коллекция слов (терминов) T . Каждый термин $t_i \in T$ имеет определенный вес w_{ij} по отношению к интернет-ресурсу $d_j \in D$ (рис. 2), т. е. встречаемость этого слова на странице интернет-ресурса. Порядок слов учитывать не будем. На основании этих признаков каждому слову ставится в соответствие его вес.

Таким образом, каждый ресурс можно представить в виде вектора весов его терминов $d_j = \{w_{1j}, \dots, w_{|T|j}\}$. Веса документов нормируем так, чтобы $w_{ij} \geq 0$ и $w_{ij} \leq 1$, где $i \in (0, |T|)$ и $j \in (0, |D|)$.

Для вычисления веса термина на странице интернет-ресурса используем классический частотный метод вычисления степени соответствия интернет-ресурса, так как этот метод относительно прост и имеет несложный алгоритм, что принципиально при обработке больших объемов документов.

Вычисляем вес термина следующим образом:

$$w_{ij} = tf_{ij} \cdot df_i, \quad (4)$$

где tf_{ij} — это отношение числа терминов t_i на странице интернет-ресурса d_j к общему числу терминов в этом документе, т. е.

$$tf_{ij} = \frac{Nt_{ij}}{|T_i|}, \quad (5)$$

где df_i — число, обратное числу интернет-ресурсов, в которых встречается термин t_i :

$$df_i = \frac{|D|}{Nd_j}, \quad (6)$$

где Nd_j — число интернет-ресурсов, в которых встретился j -й термин.

Таким образом, чем чаще термин встречается на странице интернет-ресурса, но реже встречается

ся во всех интернет-ресурсах, тем выше будет его вес в данном интернет-ресурсе.

Наиболее трудоемкой частью реализации системы является разработка процесса, отвечающего за автоматическую категоризацию интернет-

ресурсов, обучение системы на уже категоризированных интернет-ресурсах и определение соответствия категории.

Алгоритм автоматической категоризации интернет-ресурсов изображен на рис. 3 и заключается в следующем:

- вычисляется мера близости страницы интернет-ресурса и категории — степень соответствия ресурса категории;
- для каждой страницы выбирается категория, наиболее близкая к ресурсу;
- в случае если значение степени соответствия ресурса превышает некоторое пороговое значение категории, ресурс добавляется в категорию;
- в случае если значение степени соответствия ресурса не превысило порогового значения категории, ресурс не добавляется в категорию и

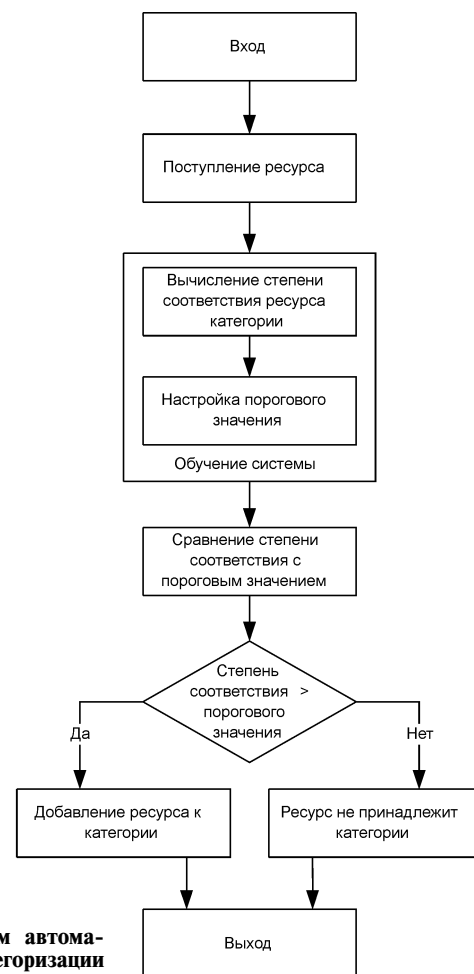


Рис. 3. Алгоритм автоматической категоризации интернет-ресурсов

решение о принадлежности его к категории определяет эксперт.

Степень соответствия (CSV) между категорией c и интернет-ресурсом d_j определяем как скалярное произведение между их векторными представлениями:

$$CSV(c, d_j) = cd_j = \sum_i c_i d_{ij}. \quad (7)$$

Будем принимать решение о принадлежности интернет-ресурса к категории, если степень соответствия достигнет заданного порога τ . Таким образом, получаем

$$\Phi'(c, d_j) = \begin{cases} 1, & CSV(c, d_j) \geq \tau \\ 0, & CSV(c, d_j) < \tau. \end{cases} \quad (8)$$

После того как вычислена степень соответствия между категорией и интернет-ресурсом, а также пороговое значение категории, необходимо провести обучение. Цель обучения — настройка весовых коэффициентов и порогового значения таким образом, чтобы процедура категоризации относилась положительные примеры к категории, а отрицательные примеры не относилась, т. е. чтобы суммы весовых коэффициентов всех положительных примеров были равны либо превышали пороговое значение, а суммы для отрицательных примеров были ниже порога.

Для оценки качества категоризации интернет-ресурсов применяем метрики из информационного поиска, такие как полнота, точность, F -мера [5].

Пусть D_r — множество интернет-ресурсов, категоризированных экспертами, а D_a — множество интернет-ресурсов, категоризированных автоматически.

Полнота категоризации интернет-ресурсов по категории вычисляется как отношение числа правильно категоризированных интернет-ресурсов системой к общему числу интернет-ресурсов, относящихся к этой категории:

$$r = \frac{|D_a \cap D_r|}{|D_r|}. \quad (9)$$

Точность категоризации интернет-ресурсов по категории вычисляется как отношение числа интернет-ресурсов, правильно категоризированных системой, к общему числу интернет-ресурсов, автоматически категоризированных системой:

$$p = \frac{|D_a \cap D_r|}{|D_a|}. \quad (10)$$

Для идеального алгоритма полнота и точность должны быть равны 100 %.

F -мера, т. е. сводная оценка качества категоризации, определяется как гармоническое среднее полноты и точности:

$$F = \frac{2rp}{r+p}. \quad (11)$$

Основная структура разрабатываемой системы высокорелевантного поиска, анализа и категоризации интернет-ресурсов и взаимодействие ее подсистем (ПС) выглядит следующим образом (рис. 4):

Подсистема "Поиск новых ресурсов" предназначена для поиска новых интернет-ресурсов. Результатом ее деятельности является набор новых адресов сайтов, пополняющих базу тематической категоризации.

Далее вся информация о новых ресурсах поступает в подсистему "Ведение информационных ресурсов". При этом есть только базовая информация о ресурсе, он не привязан к категориям.

После этого в работу включается подсистема "Обход интернет-ресурсов". В рамках данной подсистемы осуществляется обход сайта и получение набора страниц, которые можно анализировать тематически.

Следующим является подсистема категоризации интернет-ресурсов, которая анализирует тексты страниц, составляет их профиль и на основании этого принимает решение об отнесении сайта к той или иной тематической категории.

Процессом, контролирующим качество классификации, управляет подсистема "Контроль и настройка процедуры классификации".

В результате классификации в рамках подсистемы "Ведение информационных ресурсов" сайты получают соответствие категориям.

Дополнительно ведется специализированный журнал отслеживания изменений о сайтах и категориях, который используется подсистемой "Об-

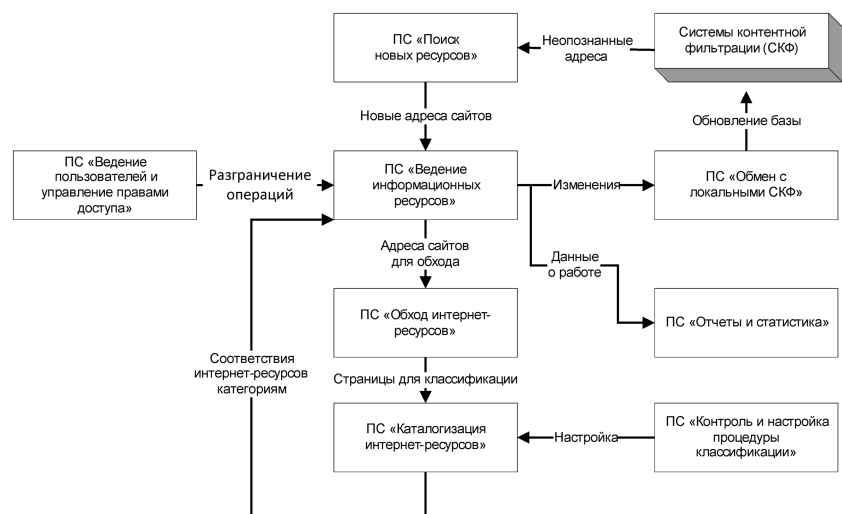


Рис. 4. Структура системы и взаимодействие ее подсистем

мен с системами контентной фильтрации (СКФ)" для обновления данных в базах СКФ и получения от них новых неизвестных адресов для анализа.

Подсистема "Ведение пользователей и управленческие правами доступа" позволяет использовать систему в многопользовательском режиме, с разграничением прав доступа между пользователями и контролем действий, выполняемых пользователем.

Подсистема "Отчеты и статистика" собирает информацию от всех модулей и предоставляет ее для анализа.

Разрабатываемая система позволит выполнять поиск новых ресурсов в сети Интернет, анализировать текстовое содержимое этих ресурсов, актуализировать накопленную информацию в соответствии с заданным списком, состоящим из 48 категорий, которые охватывают более 200 тем.

Список литературы

1. Абсалямов А. Борьба с киберслэкингом // Windows 2000 Magazine. 2000. № 3.
2. Плешко В. В., Ермаков А. Е., Голенков В. П. RCO на РОМИП 2004 // Материалы Российского семинара по оценке

методов информационного поиска (РОМИП 2004). Пушкино. 2004. С. 43—61.

3. Плешко В. В., Ермаков А. Е., Митюхин В. А. RCO на РОМИП 2003: Отчет об участии в семинаре по оценке методов информационного поиска // Труды Первого российского семинара по оценке методов информационного поиска / Под ред. И. С. Некрестьянова. СПб.: Изд. НИИ Химии СПбГУ, 2003. С. 42—51.

4. Поляков И. Е. Опыт создания системы фильтрации агрессивного web-контента // Труды XII Всероссийской научно-методической конференции "Телематика 2005", 6—9 июня 2005 г. СПб.: Изд-во СПбГУ, 2005.

5. Поляков П. Ю., Плешко В. В. RCO на РОМИП 2006 // Труды Четвертого российского семинара по оценке методов информационного поиска. СПб.: Изд. НИИ Химии СПбГУ. 2003. С. 72—79.

6. Фонд "Общественное мнение". <http://www.fom.ru/>.

7. Sebastiani F. Machine Learning in Automated Text Categorization. <http://nmis.isti.cnr.it/sebastiani/>.

8. Russian Context Optimizer. Технологии анализа и поиска текстовой информации. <http://www.rco.ru/>.

9. Некрестьянов И. С., Павлова Е. Ю. Обнаружение структурного подобия HTML-документов // Труды четвертой всероссийской конференции RCDL'2002, 38—54, Дубна, Россия, 2002.

10. Ziv Bar-Yosef, Sridhar Rajagopalan Template Detection via Data Mining and its Applications // In Proceedings of WWW2002. May 7—11. 2002. Honolulu, Hawaii, USA.

11. Gupta S., Kaiser G., Grimm P., Chiang M., Starren J. Automating Content Extraction of HTML Documents // World Wide Web Journal. January 2005.

УДК 004.738.52

А. А. Чеснавский,

Московский государственный
инженерно-физический институт
(государственный университет)

Семантическое отслеживание изменений на веб-сайтах

Рассматривается алгоритм семантического отслеживания изменения в документах HTML. Основные преимущества предложенного алгоритма — отслеживание изменений в данных веб-страницы, а не презентационной части; отсутствие необходимости знать внутреннюю структуру HTML-документа и проводить преобработку. Алгоритм может быть использован при анализе объемных веб-страниц, а также для эффективной веб-интеграции (веб-клиппинг).

Данные на многих сайтах изменяются с высокой скоростью. Где-то это дни, а иногда минуты и даже секунды. В ряде случаев необходимо обеспечить обнаружение изменений на веб-страницах. Например, биржевому брокеру необходима информация об изменении цены на определенные акции. Ручной поиск изменений может быть

времяемким и неэффективным, тогда как автоматический поиск изменений предоставит всю необходимую информацию. Еще одним примером может быть поиск информации о новых предложениях по лоту на аукционе.

В настоящее время есть довольно большое число алгоритмов обнаружения изменений (в том числе и для иерархических данных), таких как MH-DIFF, xmdiff, ATBE, TreeDiff и ряд алгоритмов для поиска изменения в плоских данных, которые обычно отображают изменения в терминах строк и столбцов исходного документа. То есть обычно результатом работы подобных алгоритмов является сообщение вида: "Строка N в документе A отличается от строки N в документе B", что на самом деле не является интуитивно понятным результатом в контексте иерархических данных и документов HTML.

Такие алгоритмы как MH-DIFF и TreeDiff генерируют более значимые результаты для иерархических данных, нежели их аналоги для анализа изменений в плоских документах, и могут выглядеть следующим образом: "Значение узла A было изменено на B". Однако эти алгоритмы, к сожалению, не применимы для HTML-документов ввиду того, что данные и отображение данных являются частями одного и того же документа.

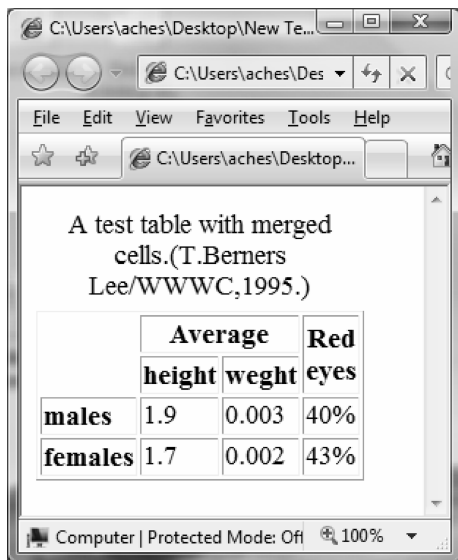


Рис. 1. Пример таблицы

Рассмотрим пример HTML-документа [1]:

```
< 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><THALIGN = left > females < TD > 1.7 < TD > 0.002 < td > 43 %
</TABLE >
```

Предположим (рис. 1), что значение 40 % в последней колонке изменяется на 50 %. В данном примере это означает, что значение процента красных глаз у мужчин изменилось с 40 на 50 %. Для того чтобы технически осуществить данное изменение нужно в HTML-коде в строке 5 заменить 40 % на 50 %. Как уже говорилось выше, при анализе изменений сообщение об изменении в строке 5 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 мужчин. Подобные проблемы могут быть успешно разрешены с помощью алгоритма семантического отслеживания изменений (АСОИ).

АСОИ отслеживает изменения, которые мы назовем семантическими изменениями, в HTML-документах в рамках иерархии данных в отличие от иерархии разметки HTML-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 отслежи-

вания 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 теги у некоторых HTML-элементов могут отсутствовать или быть необязательными, что приводит к сложностям в разборе (парсинге) HTML-документов. В результате, вследствие этих отличий получение информации из HTML-документов требует дополнительных знаний о внутренней структуре или предобработки исходных документов [2] (что в реальной практике может быть недоступно), чего не нужно для работы АСОИ.

Итак, иерархически структурированные данные — это набор данных D , в котором каждый неделимый элемент формирует узел в соответствующем дереве T , где T представляет данные иерархии D . Полуструктурированные данные, данные в HTML/XML-документах и такие логические сущности, как файлы и папки, являются типичными примерами иерархических данных. Мы представляем иерархические данные как ориентированное дерево, где есть дуга от узла v_2 к узлу v_1 , если v_1 — это прямой контейнер v_2 в иерархии данных. Этот тип отношений может быть легко найден в любом элементе контейнерного типа в HTML/XML или структуре папка — подпапка/файл в файловой системе. Рассмотрим понятие ветви в иерархии данных.

Определение 1. Допустим, есть ориентированное дерево T . Ветвью T является путь от листового узла v_n к корневой вершине v_1 дерева T , обозначаемой $.v_1.v_2...v_n$ ($n \geq 1$), где каждый узел отделен точкой от смежного узла и v_i является предком v_{i+1} ($1 \leq i \leq n-1$). Любой подпуть ветви, который включает листовую вершину, т. е. $v_i.v_{i+1}...v_n$ ($1 \leq i \leq n$) называется частичной ветвью в дереве T и обозначается без начальной точки. Кроме того, для данной ветви $.v_1...v_n$, $.v_1...v_{i-1}$ называется контекстом v_i ($1 \leq i \leq n$).

Запись $.A.(A_1, A_2, ..., A_n)$, называемую составной ветвью, обозначим $(.A. A_1, A. A_2, ..., A. A_n)$, где каждая $.A. A_i$ ($1 \leq i \leq n$) является ветвью. Более того, если существует более одного узла от одного родительского узла o ("братья"), мы будем каждый узел отделять один от другого суффиксом, т. е. $o[1]$, $o[2]$ и т. д., согласно порядку их появления сверху вниз и слева направо в исходном документе.

АСОИ состоит из трех шагов:

1) конструирование семантических иерархий (деревьев) любых двух данных HTML-документов;

2) идентификация ветвей в результирующих деревьях и удаление идентичных;

3) определение изменений одного дерева относительно другого на основе сравнения каждой из оставшихся ветвей.

HTML-документ состоит из одного или более логических секций которые:

- являются равными друг другу, например Section 1, Section 2, и т. д.;
- одна секция структурно включает другую, например Section 1 и Section 1.2;
- две секции не являются равными друг другу, и одна из них не включает другую, например Section 1.3 и Section 4.

Наша задача — определить семантическую иерархию секций в HTML-документе, используя различные HTML-теги. Как уже было сказано выше, HTML был создан не только для определения, но и для отображения данных и, таким образом, большинство HTML-документов не способствуют организации компонентов HTML в секции или блоки согласно иерархии. Таким образом, более детально, наша первая задача заключается в идентификации, какие HTML-теги могут быть использованы для конструирования иерархической структуры HTML-документов (тип 1), а которые служат для представления данных (тип 2). Список тегов с разделением по типам можно найти в табл. 1.

Конструирование семантической иерархии для нетабличных данных состоит из двух шагов. На первом шаге все теги типа 2 удаляются из исходного HTML-документа. Отметим, что удаление тегов 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 тега `< i >`. На втором шаге семантическая иерархия конструируется на основе предшествования нетабличных HTML-тегов так, как это изображено на рис. 2. Предшествование между двумя HTML-элементами А и В, обозначаемое $A \gg B$ показывает, что данные, содержащиеся в А, выше в соответствующей иерархии, чем данные, содержащиеся в В.

Определив порядок предшествования среди тегов типа 1 (за исключением тегов, предназначенных для создания таблиц) в HTML-документе Н, мы применяем следующие правила к синтаксической иерархии Н для конструирования семантической иерархии S данных в Н.

1. Создать корневой узел V_r иерархии S из #PCDATA из эле-

Группы HTML-тегов

HTML теги		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		ADDRESS		

мента TITLE. Элемент TITLE, который является обязательным для любого HTML-документа согласно спецификации HTML, содержит описание документа.

2. Создать иерархию KEYWORDS \leftarrow "список ключевых слов" из любого тега META в виде `< META NAME = "keywords" VALUE = "list_of_keywords" >`. Присоединить иерархию к V_r и получить $V_r \leftarrow$ KEYWORDS \leftarrow "список ключевых слов". Вообще, META является опциональным и предоставляет произвольное число ключевых слов, каждое из которых релевантно исходному документу.

3. Создать иерархию ADDRESS \leftarrow text1 text2 text3 \leftarrow HREF = "link", если элемент ADDRESS в

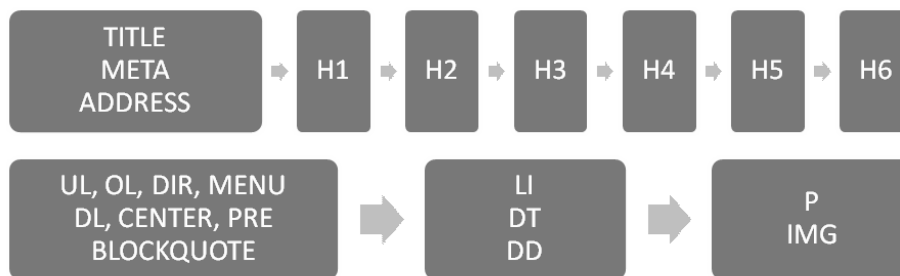


Рис. 2. Порядок предшествования нетабличных элементов (тип 1)

форме $\langle \text{ADDRESS} \rangle \text{ text } 1 \langle \text{A HREF} = \text{"link"} \rangle \text{ text } 2 \langle /\text{a} \rangle \text{ text } 3 \langle /\text{address} \rangle$, где $\text{text } 1$ и $\text{text } 3$ могут быть пустыми. Создать иерархию $\text{ADDRESS} \leftarrow \text{text}$ вместо предыдущего варианта, если элемент ADDRESS в форме $\langle \text{ADDRESS} \rangle \text{ text} \langle /\text{ADDRESS} \rangle$, и присоединить к иерархии V_r . Отметим, что элемент ADDRESS является опциональным и, если он существует, он должен быть потомком элемента BODY .

4. Для всех оставшихся элементов HTML-документа применять следующее. Для HTML-элемента в форме $\langle \text{t1} \rangle \# \text{PCDATA} \langle \text{t2} \rangle \dots \langle /\text{t2} \rangle \langle /\text{t1} \rangle$, где t1 (t2 соответственно) — название элемента HTML, создать иерархию $\# \text{PCDATA} \leftarrow \text{sh}$, где sh — семантическая иерархия, полученная из $\langle \text{t2} \rangle \dots \langle /\text{t2} \rangle$. Когда после $\# \text{PCDATA}$ идут другие HTML-теги, это обычно свидетельствует о том, что данные включены в теги. Отсюда мы полагаем, что $\# \text{PCDATA}$ содержит данные. Данные ветви в составном виде:

$\dots l_0.(l_1\dots d_1, l_2\dots d_2, \dots, l_n\dots d_n)$, если $d_1\dots d_n$ являются $\# \text{PCDATA}$;

$\dots l_0.(l_1\dots d_1, l_2\dots d_2, \dots, l_n\dots d_n\dots)$, если d_n — единственная таблица в $l_n\dots d_n$, где l_i ($0 \leq i < n$) — элемент HTML. Создаем иерархию $d_i \leftarrow d_n$ ($1 \leq i \leq n$), если $l_i \gg l_n$ и $l_k \leftarrow l_n$, для любого k ($i + 1 \leq k < n$). Присоединить дугу от корня для каждой сконструированной иерархии к V_r .

Это правило говорит о том, что для данных (d_i), заключенных в d_n , мы создаем иерархическую связь между ними. Если d_n является таблицей, то используется заглушка TABLE in $d_i \leftarrow \text{TABLE}$, которая затем будет заменена на корневой элемент семантической иерархии соответствующей таблицы. Условие, что d_n является единственной таблицей в $l_n\dots d_n$, подразумевает, что эта таблица есть внешняя таблица в этой ветви.

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. Будем в дальнейшем 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 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, рассмотренный выше (см. рис. 1). В данном примере наблюдается различное число строк и столбцов, что затрудняет процесс корреляции строк и столбцов. Когда TH или TD включает $\text{ROWSPAN} = \text{"n"}$ ($\text{COLSPAN} = \text{"n"}$ 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 теги 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.

Определение 2. Псевдотаблицей $T = \{(a_{11}, \dots, a_{1n}), (a_{21}, \dots, a_{2n}), \dots, (a_{m1}, \dots, a_{mn})\}$ со столбцами-заголовками C_1, \dots, C_n и заглавием (caption) C является двумерная таблица, где заголовок, каждый столбец-заголовок и данные ячейки a_{ij} ($1 \leq i \leq m$, $1 \leq j \leq n$) могут быть нулевыми. Кроме того, $a_{k1} \neq a_{l1}$, если $k \neq l$ ($1 \leq k, l \leq m$).

Согласно спецификации HTML-таблица содержит, по крайней мере, один элемент CAPTION . Если CAPTION определен для HTML-таблицы, он становится заголовком C соответствующей псевдотаблицы. В противном случае, мы присваиваем заголовку значение "TABLE", служащее заглушкой, которая в итоге будет удалена.

HTML-грамматика определяет иерархию HTML-документа отношением контейнер/содержимое между тегами и данными, что отлично от иерархии в псевдотаблице, поскольку в псевдотаблице нет тегов. Поскольку столбцы-заголовки и табличные данные могут быть нулевыми в псевдотаблице, рассмотрим особый тип отношений включения: в отношении контейнер — содержимое $o_1 \leftarrow o_2 \leftarrow o_3$, где o_i ($1 \leq i \leq 3$) является либо заголовком, либо данными, отношение может быть редуцировано до $o_1 \leftarrow o_3$, если o_2 является пустой строкой.

C		
C ₁	...	C _n
A _{1,1}	...	A _{1,n}
...
A _{m,1}	...	A _{m,n}

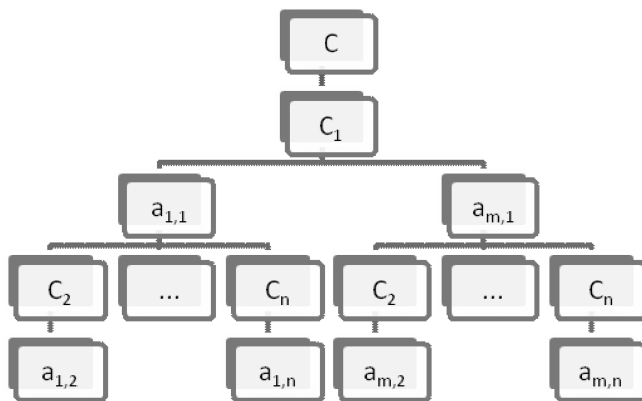


Рис. 3. Псевдотаблица T и соответствующая семантическая иерархия SH_t

Определение 3. N-арная псевдотаблица $T = \{(a_{11}, \dots, a_{1n}), (a_{21}, \dots, a_{2n}), \dots, (a_{m1}, \dots, a_{mn})\}$ со столбцами-заголовками C_1, \dots, C_n и заглавием C, семантическая иерархия $SH = (V, E, g)$ от T, обозначенная SH_t , есть ориентированное дерево, где $V_r \in V$ является корневой вершиной SH_t , обозначенной C, и каждый узел $v \in V$, отличный от V_r , обозначает непустой a_{ij} (C_j соответственно) ($1 \leq i \leq m, 1 \leq j \leq n$) в T и помечен a_{ij} (C_j соответственно); E — конечное множество ориентированных дуг; $g: E \rightarrow V^*V$ — функция, такая, что $v_2 \leftarrow v_1$, если $g(e) = (v_1, v_2)$ и $V_r(c) \leftarrow C_1 \leftarrow a_{i1} \leftarrow C_j \leftarrow a_{ij}$ ($1 \leq i \leq m, 2 \leq j \leq n$).

Так как заглавие псевдотаблицы T содержит краткое описание того, для чего создана таблица, заглавие выбрано в качестве корневой вершины соответствующего SH_t . В определении семантической иерархии SH_t содержит поддеревья, имеющие корневыми вершинами $a_{i1} \dots a_{in}$ ($1 \leq i \leq m$) с ограничением $V_r(c) \leftarrow C_1 \leftarrow a_{i1}$ ($1 \leq i \leq m$). Это связано с тем, что строка может быть уникально идентифицирована от других строк по первому столбцу (т. е. a_{i1}) в T (см. определение 2). На рис. 3 показаны псевдотаблица и соответствующая семантическая иерархия, сделанная на основе оп-

ределения 3. Основная задача в конструировании псевдотаблицы — определить каждую строку, т. е. $a_{i1} \dots a_{in}$ ($1 \leq i \leq m$) и столбец, т. е. $a_{1j} \dots a_{mj}$ ($1 \leq j \leq n$) из соответствующей HTML-таблицы.

Рис. 4 иллюстрирует, как можно преобразовать данные в элементах TH и TD в столбцовой таблице HTML в C_j и a_{ij} псевдотаблицы. Рис. 5 и рис. 6 показывают преобразование строчной и строчно-столбцовой таблиц HTML соответственно в псевдотаблицы.

Как уже говорилось выше, HTML-таблица может иметь различное число столбцов в строках вследствие использования атрибутов COLSPAN и ROWSPAN. Если элемент TH или TD содержит COLSPAN = "n", соответствующая ячейка TH или TD расширяется на n столбцов и занимает, таким образом, n ячеек, включая текущую ячейку в текущей строке. Таким образом, можно считать, что вставляются n - 1 ячеек вправо от текущей ячейки и в них реплицируются данные текущей ячейки. ROWSPAN функционирует иначе. Если элемент TH содержит ROWSPAN = "n", конкретная ячейка расширяется на следующие n - 1 строк и занимает n ячеек. В этом случае мы вставляем n - 1 ячеек ниже текущей ячейки и не реплицируем содержимое текущей ячейки h в каждую из вставленных ячеек, а только записываем h во вставленную N - 1 ячейку. Таким образом, h появляется только в ячейке n - 1, все остальные вставленные ячейки остаются пустыми. Это необходимо для сохранения корректных взаимосвязей табличных данных по всем строкам в столбцах, избегая повторения одного и того же заголовка, так как конкатенированные заголовки в столбце

h _{1,1}	...	h _{1,n}
...
h _{k,1}	...	h _{k,n}
d _{k+1,1}	...	d _{k+1,n}
...
d _{k+m,1}	...	d _{k+m,n}

C ₁ =h _{1,1} ...h _{k,1}	...	C _n =h _{1,n} ...h _{k,n}
A _{1,1} =d _{k+1,1}	...	A _{1,n} =d _{k+1,n}
...
A _{m,1} =d _{k+m,1}	...	A _{m,n} =d _{k+m,n}

Рис. 4. Перевод столбцовой таблицы HTML в соответствующую псевдотаблицу

h _{1,1}	...	h _{1,l}	d _{1,l+1}	...	d _{1,l+n-1}
...
h _{m,1}	...	h _{m,l}	d _{m,l+1}	...	d _{m,l+n-1}

A _{1,1} =h _{1,1} ...h _{1,l}	A _{1,2} =d _{1,l+1}	...	A _{1,n} =d _{1,l+n-1}
...
A _{m,1} =h _{m,1} ...h _{m,l}	A _{m,2} =d _{m,l+1}	...	A _{m,n} =d _{m,l+n-1}

Рис. 5. Перевод строчной таблицы HTML в соответствующую псевдотаблицу

hh _{1,1}		hh _{1,2}		...	hh _{1,n}
...	
hh _{k,1}		hh _{k,2}		...	hh _{k,n}
hv _{k+1,1}	...	hv _{k+1,l}	d _{k+1,l+1}	...	d _{k+1,l+n-1}
...
hv _{k+m,1}	...	hv _{k+m,l}	d _{k+m,l+1}	...	d _{k+m,l+n}
1		l	1		-1

C ₁ =hh _{1,1} ...hh _{k,1}	C ₂ =hh _{1,2} ...hh _{k,2}	...	C _n =hh _{1,n} ...hh _{k,n}
A _{1,1} =hv _{k+1,1} ...hv _{k+1,l}	A _{1,2} =d _{k+1,l+1}	...	A _{1,n} =d _{k+1,l+n-1}
...
A _{m,1} =hv _{k+m,1} ...hv _{k+m,l}	A _{m,2} =d _{k+m,l+1}	...	A _{m,n} =d _{k+m,l+n-1}
l			

Рис. 6. Перевод строчно-столбцовой таблицы HTML в соответствующую псевдотаблицу

HTML-таблицы преобразуются в заголовок столбца псевдотаблицы. Однако, если ROWSPAN содержится в TD, мы вставляем $n - 1$ новых ячеек ниже текущей ячейки TD, и реплицируем в них данные текущей ячейки для того, чтобы данные в каждой из n различных строк одного и того же столбца были одинаковыми. После того, как обработка COLSPAN и ROWSPAN прошла успешно, результирующая таблица удовлетворяет определению псевдотаблицы.

Рассмотрим случаи вложенных таблиц. Если таблица T_2 вложена в другую таблицу T_1 , T_2 находится в элементе TD таблицы T_1 . Пусть S — семантическая иерархия T_2 и пусть td будет элементом в T_1 , который содержит T_2 . Тогда, создадим $d \leftarrow S$, если d предшествует #PCDATA в td . В противном случае S обрабатывается, как #PCDATA в td . Для конструирования семантической иерархии всего HTML-документа H , включающего в себя таблицу T , мы присоединяем дуги семантической иерархии T к семантической иерархии H по следующему правилу.

Дана иерархия $d \leftarrow TABLE$, полученная по правилу 4, и семантическая иерархия S таблицы T , связанной с $TABLE$. Заменяем $d \leftarrow TABLE$ так, что:

$d \leftarrow S$, если S представлена в виде C . (...) или $d \leftarrow c_1, \dots, d \leftarrow c_n$, если S представлена в виде $TABLE.(c_1 \dots, \dots, c_n \dots)$,

где C — CAPTION в T и $c_1 \dots c_n$ — табличные данные T .

Если HTML-документ в теле BODY не имеет ничего кроме таблицы, исходящая дуга от семантической иерархии таблицы проходит к содержанию элемента TITLE документа H .

Теперь пришло время рассмотреть обнаруженные изменения между двумя семантическими иерархиями, т. е. между двумя HTML-документами. В качестве примера рассмотрим две семантические иерархии S_1 и S_2 (рис. 7 и рис. 8). За один обход дерева могут быть определены ветви дерева. Таким образом, ветви S_1 и S_2 идентифицированы (X_i ($1 \leq i \leq 5$) и Y_j ($1 \leq j \leq 7$)):

$S_1 = \{x_1 = .a.b.f, x_2 = .a.c.k, x_3 = .a.c.g, x_4 = .a.d.h, x_5 = .a.e.p\}$;

$S_2 = \{y_1 = .a.g.b, y_2 = .a.c.k, y_3 = .a.c.g.b.f, y_4 = .a.d.c, y_5 = .a.h.d.h, y_6 = .a.h.e, y_7 = .a.h.p\}$.

При определении отличий между двумя семантическими иерархиями S_1 и S_2 в первую очередь удалить ветви, имеющиеся в обеих иерархиях. После этого основная задача — определить какие из оставшихся ветвей в S_1 "скорее всего" изменились и стали ветвями в S_2 .

Пусть даны исходные ветви b_1, \dots, b_m и целевая ветвь b_t . Рассмотрим критерии того, что ветвь b_i ($1 \leq i \leq m$) ближе других находится в b_t .

1. Если для b_i нужно "меньше изменений", чтобы прийти к b_t , чем другим ветвям. В этом случае мы полагаем b_i источником b_t .

2. В случае, когда любые две или более ветвей b_1, b_2, \dots, b_m требуют тех же затрат для модификации в b_t , b_i выбирается, если b_i и b_t имеют большее число совпадающих вершин по направлению к вершине, чем другие b_j ($1 \leq j \leq m, i \neq j$).

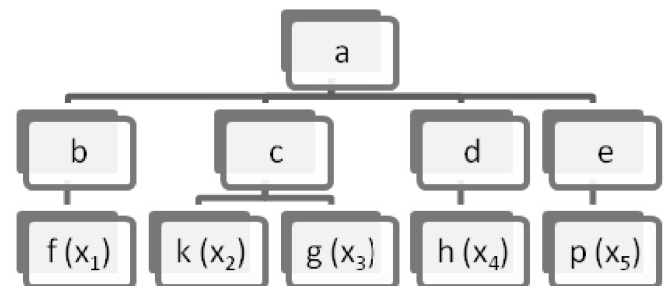


Рис. 7. Дерево S1

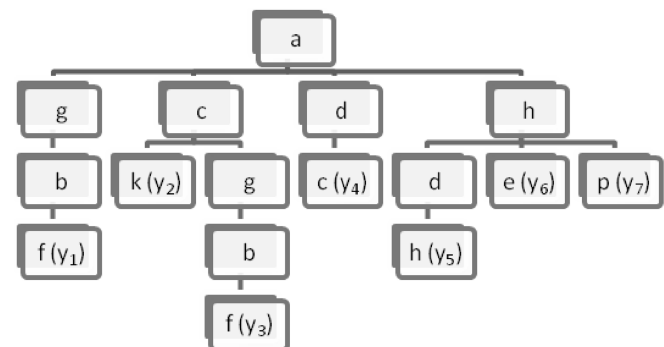


Рис. 8. Дерево S2

Введем понятие взвешенной разницы δ для измерения значения отличия (т. е. изменений) между двумя ветвями. С использованием δ фиксируем следующие отличия.

1. Отличия около корневых вершин в b_1 и b_2 более значимы, нежели отличия около листовых вершин. Это связано с тем, что корень иерархии данных определяет частный класс данных, в то время как листовые вершины иерархии данных — это элемент данных, который принадлежит классу или подклассу корня.

Рассмотрим пример: $b_1 = \text{НазваниеКниги. 'Мертвые души'}$, $b_2 = \text{НазваниеКниги. 'Война и Мир'}$, $b_3 = \text{НазваниеФильма. 'Весна на Заречной улице'}$. Все указанные ветви различны, но мы считаем, что b_1 "более отлична" от b_3 , нежели от b_2 , так как b_1 и b_2 содержат названия книг, а b_3 — название фильма.

2. Порядок между вершинами в b_1 и b_2 является существенным. Если b_1 и b_2 имеют идентичный набор вершин, но они по-разному упорядочены, эти ветви считаются различными.

Определение 4. Пусть дана ветвь $b = .v_1 \dots v_n$. Вес вершины $v_i = (2^i)^{-1}$ ($1 \leq i \leq n$).

Для вычисления взвешенной разницы между любыми двумя ветвями, будем рассматривать каждую ветвь как упорядоченное множество с дубликатами, т. е. множество, которое может содержать повторяющиеся элементы и упорядоченность между элементами имеет значение. Например, ветвь $.a.b.c.d$ представляется как $\{a, b, c, d\}$ и является отличной от $\{a, b, c, d, b\}$. Для двух множеств с дубликатами $M_1 = \{a_1, \dots, a_m\}$ и $M_2 = \{b_1, \dots, b_n\}$ определим следующие операции:

$M_1 \cap M_2 = e_1, \dots, e_n$, где каждый e_i принадлежит как M_1 , так и M_2 и для любого i , $1 \leq i \leq n$, e_i предшествует e_{i+1} в M_1 и M_2 .

$$M_1 - M_2 = \{e | e \in M_1 \text{ и } e \notin M_1 \cap M_2\}$$

Определение 5. Пусть даны две ветви $b_1 = .u_1 \dots u_m$ и $b_2 = .v_1 \dots v_n$, взвешенная разница δ для b_1 и b_2 следующая:

$\delta(b_1, b_2) = \sum_i w_{u_i} + \sum_j w_{v_j}$ при условии, что $u_i \in (M_{b_1} - M_{b_2})$ и $v_j \in (M_{b_2} - M_{b_1})$, где w_{u_i} (w_{v_j}) — вес u_i (v_j соответственно) и M_{b_1} (M_{b_2}) — упорядоченное множество с дубликатами M_{b_1} (M_{b_2} соответственно).

Пример. Рассмотрим деревья S_1 и S_2 . $\delta(x_1, y_1) = 0,25$, тогда как $M_{x_1} - M_{y_1} = \emptyset$, $M_{y_1} - M_{x_1} = \{g\}$ и вес $g = (2^2)^{-1}$.

На третьей стадии происходит сравнение оставшихся ветвей в S_1 с каждой оставшейся ветвью S_2 на основе вычисления взвешенной разницы. Если ветвь x в S_1 "наименее отлична" от y в S_2 , тогда считаем, что ветвь y есть измененная x . Если несколько ветвей имеют одинаковые характеристики отличия — можно выбрать любую из них.

В данной статье был предложен алгоритм семантического отслеживания изменений (АСОИ), который позволяет выявить изменения данных в теле HTML-документа, а не изменения разметки документа. Особенностью данного алгоритма является то, что не требуется проводить преобразование документа и знать внутреннюю структуру HTML-страницы. АСОИ может быть использован либо как самостоятельный инструмент анализа изменений в больших документах, либо стать компонентом семантического анализа изменений HTML-страниц для подсистемы веб-клиппинга системы интеграции приложений.

Список литературы

1. **Raggett D., Hors A. L., Jacobs I.** HTML 4.0 Specification-W3C Recommendation. <http://www.w3.org/TR/REC/html40>. April 1998.
2. **Atzeni P., Mecca G.** Cut and Paste // Proc. of the 16th Intl. Symposium on Principles of Database Systems. May 1997. P. 144—153.



УДК 685.324

Д. Л. Жусов,

Академия ФСО России, г. Орел

Алгоритм обнаружения компьютерных атак на web-сервер

Предложен алгоритм обнаружения компьютерных атак на web-серверы с динамически формируемыми страницами, позволяющий повысить защищенность web-серверов.

Введение

Развитие web-технологий определило активное использование динамического метода формирования web-страниц. В отличие от статического метода, при котором заранее создается ограниченное число web-страниц, динамический метод основан на применении прикладного программного обеспечения (ПО), позволяющего формировать страницы, содержание которых зависит от значений параметров доступа запросов. При динамическом создании web-страниц клиентский запрос, структура которого определена в [1, 2], после проверки на корректность базовым ПО сервера передается прикладному ПО, которое возвращает результат обработки запроса. В отличие от атак на базовое ПО, основанных на использовании его недокументированных возможностей, причиной успешной реализации компьютерных атак на прикладное ПО является недостаточная проверка данных.

В связи с этим для обеспечения безопасного функционирования web-серверов необходимо разработать алгоритм обнаружения компьютерных атак, позволяющий верифицировать поступающие для обработки сервером данные клиентов на предмет их безопасности.

Результаты исследования

В настоящее время для идентификации факта компьютерной атаки в общем случае применяются методы, основанные на использовании сигнатур компьютерной атаки [3]. Однако эффективность данного подхода напрямую связана с обязательным постоянным пополнением базы сигнатур атак.

В работах [4, 5] предложена модель обнаружения компьютерных атак на web-сервер, функционирование которого основано на применении динамического метода формирования web-страниц, отличающаяся использованием заранее заданных безопасных запросов к серверу. Однако в данной модели *не учитывается* тот факт, что запросы, имеющие одинаковые число переменных доступа и размер значения переменных доступа, на практике могут быть принципиально различными. Например, <http://www.mail.ru?parl=http://www.yandex.ru/index.php> по этим признакам идентичен запросу <http://www.mail.ru?parl=http://www.attack.ru/atack.php>, хотя результаты его выполнения будут различными.

В связи с этим для повышения защищенности web-сервера необходим алгоритм обнаружения компьютерных атак на web-сервер, позволяющий выполнять процедуру верификации поступающего запроса множеству безопасных запросов. Данная процедура реализована на основе методов теории распознавания образов, анализ которых обусловил применение конечных автоматов. Предлагаемый подход к обнаружению компьютерных атак на web-сервер позволил разработать функциональную модель системы обнаружения компьютерных атак (рис. 1).

Процедура верификации прикладного ПО осуществляется на основе подхода, изложенного в [6]. Результатом выполнения данного этапа является множество безопасных значений параметров доступа запросов, которые затем используются в процедурах обнаружения атак.

Для обнаружения компьютерных атак (изменений в запросах) предлагается реализовать процедуру проверки соответствия значений параметров поступившего запроса безопасному множеству путем работы автомата-распознавателя, структурная схема которого представлена на рис. 2.

Этап выделения и распознавания идентификатора ресурса позволяет идентифицировать факт того, к какому из доступных ресурсов web-сервера происходит обращение пользователя.

Этап выделения и распознавания метода доступа к ресурсу позволяет проверить, что анализируемый запрос к информационному ресурсу сформирован на основе допустимого метода.

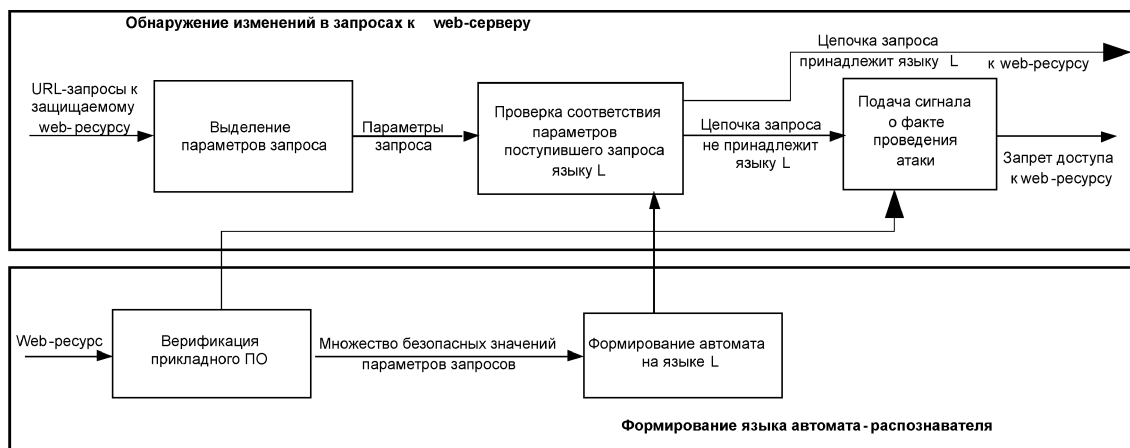


Рис. 1. Функциональная модель системы обнаружения компьютерных атак на web-сервер



Рис. 2. Структурная схема автомата-распознавателя компьютерных атак на web-сервер

В результате выделения переменных доступа и их значений формируются пары "переменная — значение", которые верифицируются с разрешенным списком.

Процедура верификации может быть представлена графом состояний и переходов (рис. 3).

Процедура распознавания переменных доступа и их значений, представленная выше, описывает распознавание одной пары "переменная доступа — значение". Для проверки всех пар необходимо, чтобы данная процедура отработала столько раз, сколько пар необходимо распознать. Состояние S_1 обозначает процесс распознавания имени переменной доступа, а S_2 — ее значения.

На основе вышеизложенного был разработан алгоритм обнаружения компьютерных атак на web-сервер (рис. 4).

В процессе работы алгоритма возможно установление флага состояния автомата-распознавателя в одно из двух положений. В случае анализа цепочки запроса, состоящей из безопасных значений параметров доступа, флаг устанавливается в положение f . При невыполнении данного условия анализ цепочки поступившего запроса оканчивается установлением флага в значение w , что означает идентификацию при анализе цепочки небезопасных значений.

Предлагаемый алгоритм позволяет провести операцию верификации между поступившими значениями переменных доступа и разрешенными из заранее сформированного списка безопасных значений, что обеспечивает обработку web-сервером только корректно сформированных запросов с безопасными значениями параметров доступа. Эффективность алгоритма определяется полнотой и качеством выполнения процедуры верификации прикладного ПО в виде формирования полного множества безопасных запросов.

В рамках исследований был разработан модуль обнаружения атак на web-сервер, реализующий представленный алгоритм. Пусть на web-странице легально присутствует ссылка <http://www.mail.ru?parl=>

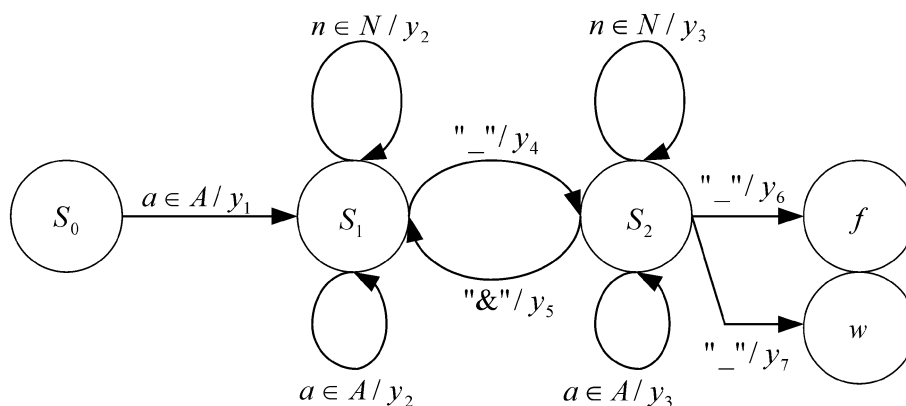


Рис. 3. Граф состояний и переходов процесса функционирования процедуры распознавания переменных доступа и их значений

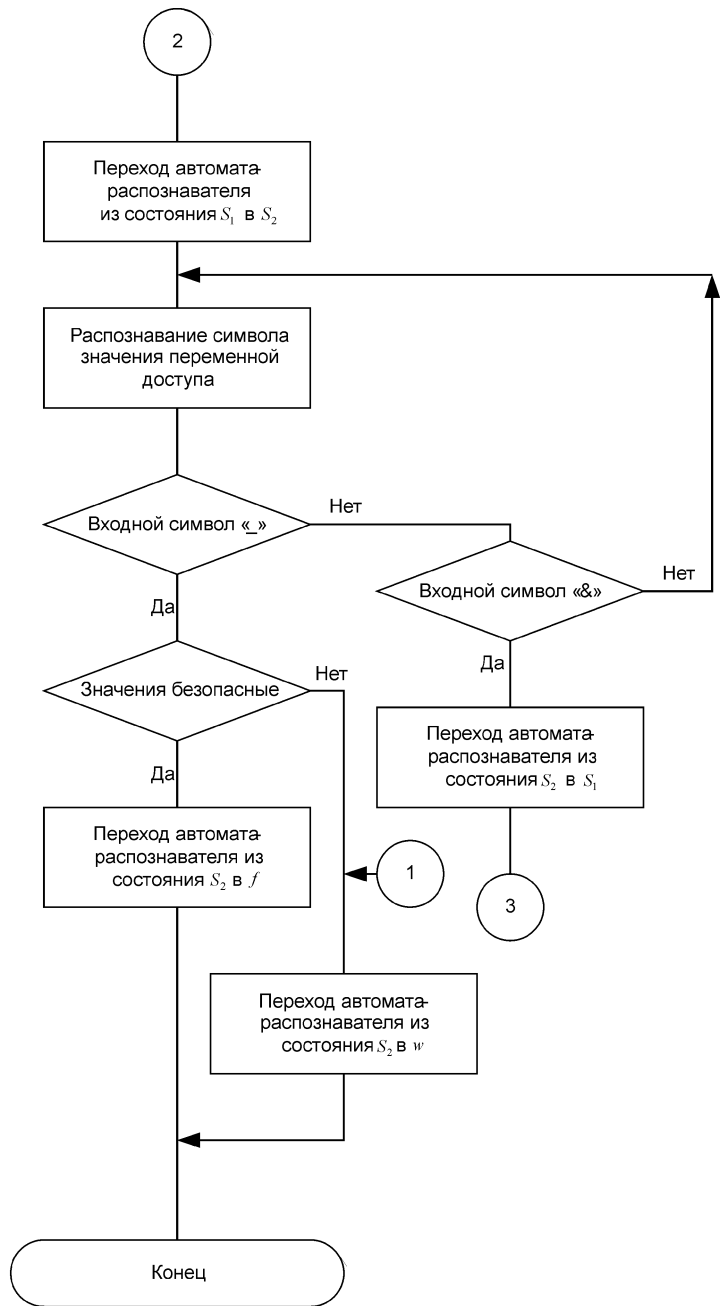
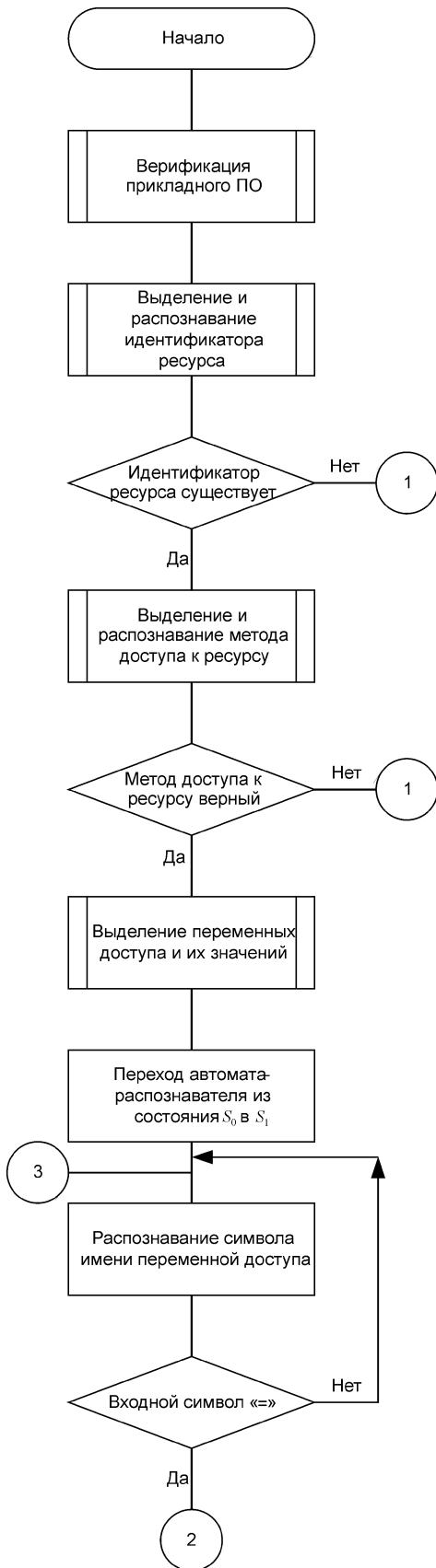


Рис. 4. Алгоритм обнаружения компьютерных атак на web-сервер

<http://www.nait.ru/index.php>. Злоумышленник в целях реализации компьютерной атаки отправляет серверу запрос `http://www.mail.ru?parl = http://www.atak.ru/atak.php`. Модуль содержит множество безопасных запросов и верифицирует поступающий запрос с множеством безопасных значений. В результате анализа поступившего запроса модулем было выявлено запрещенное значение переменной доступа `parl`, вследствие чего доступ к серверу посредством данного запроса был запрещен. В то же время, используя системы, построенные на сигнатурном методе обнаружения аномалий, например Snort, невозможно обнаружить факт модификации значений перемен-

ных доступа ввиду того, что они не содержат запрещенные комбинации символов (признаки компьютерной атаки).

Таким образом, разработанным алгоритмом представлен вариант реализации процедуры обнаружения компьютерных атак на web-сервер, позволяющий в реальном режиме работы осуществлять проверку содержимого запросов к серверу на наличие признаков компьютерной атаки, реализуемой посредством изменения параметров доступа к информационному web-ресурсу, и повысить защищенность web-сервера в целом.

Список литературы

1. **Berners-Lee T., Fielding R. and Frystyk H.** Hypertext Transfer Protocol — HTTP/1.0, RFC 1945, May 1996.

2. **Fielding R., Gettys J.** Request for Comments 2616 — Hypertext Transfer Protocol — HTTP/1.1 // Network Working Group. 1999.

3. **Лукацкий А. В.** Обнаружение атак. СПб.: БХВ-Петербург, 2001.

4. **Сердюк В. А.** Математическая модель поведенческого метода обнаружения информационных атак // Тез. Междунар. молодежной науч. конф. "XXIX Гагаринские чтения". М.: МАТИ. 2004. Т. 5. С. 5—6.

5. **Сердюк В. А.** Математическая модель поведенческого метода обнаружения сетевых атак, базирующаяся на конечных автоматах-распознавателях // Моделирование. Теория, методы и средства: Матер. IV Междунар. науч.-практ. конф., г. Новочеркасск, 9 апр. 2004 г.: В 4 ч. Юж.-Рос. гос. техн. ун-т (НИИ). Новочеркасск: ЮРГТУ, 2004. Ч. 4. С. 8—13.

6. **Бухарин В. В.** Применение верификации для специального программного обеспечения в автоматизированных системах управления. Системы связи. Анализ. Синтез. Управление / Под ред. В. П. Постюшкова. СПб.: Тема, 2005. Вып. № 15. С. 9—14.

УДК 519.856

Л. Е. Мистров,

Центральный филиал ГОУ ВПО

"Российская академия правосудия", г. Воронеж

Метод оценки эффективности применения комплексов информационной индивидуальной и групповой безопасности организационно-технических систем в конфликтной неопределенности

Предлагается метод оценки эффективности применения комплексов информационной индивидуальной и групповой безопасности для обеспечения одиночных и групповых действий организационно-технических систем в условиях конфликтной неопределенности. Метод разработан на основе принципа гарантированного результата, теорий исследования операций, максимина, оптимального распределения ресурсов и метода динамического программирования, позволяющих установить оптимальные стратегии поведения противоборствующих сторон в конфликте и выбрать оптимальные варианты состава комплексов информационной безопасности.

Для современного этапа развития социально-экономических отношений характерна тенденция возрастания конкурентной борьбы различного типа финансовых, экономических организаций за овладение, перераспределение или использова-

ние тех или иных ресурсов. Ее исход определяет условия устойчивого функционирования или гибели (поглощения) организаций, вынуждая их для обеспечения заданной (максимальной) эффективности выделять ресурсы сил и средств для ведения оборонительных действий и осуществления, в свою очередь, ответных действий для нейтрализации конкурирующей стороны до некоторого минимального уровня.

В общем случае структура любой организации представляет собой организационно-техническую систему (ОТС), состоящую из совокупности объединенных единством цели элементов (ОТС меньшего уровня, технических систем и комплексов) управления, добывания информации и исполнения в данной предметной области. Применение ОТС при реализации исполнительных действий осуществляется на основе взаимосвязанной по цели, задачам, месту, времени и ресурсу совокупности организационных, организационно-технических и технических методов, мероприятий или средств и комплексов (исполнительных элементов (ИЭ)), направленных на выполнение поставленных задач независимо от различного рода внешних воздействий. Выполнение ОТС задач осуществляется проведением различного рода операций, определяемых целевым предназначением организации, в виде совокупности одиночных и групповых действий (ОД, ГД) для подавления (нейтрализации наиболее важных элементов на время операции) или воздействия ИЭ на элементы противодействующей ОТС (для определенности ОТС {B}). Это достигается распределением ОТС имеющегося ресур-

са ИЭ (в том числе и ИЭ вспомогательных систем, выполняющих совместно с ОТС поставленные задачи) для парирования эффективности действий одной или нескольких противодействующих ОТС [1, 2].

Выполнение ОТС поставленных задач сопряжено с преодолением ее ИЭ сопротивления пространственно-распределенной эшелонированной подсистемы противодействия ОТС $\{B\}$, основу которого составляет множество различных организационных, организационно-технических и технических способов или средств и комплексов противодействия — оборонительных элементов (ОЭ), отличающихся друг от друга дальностью и эффективностью применения. Применение ОЭ основывается на использовании в их составе различного типа, количества, эффективности и условий применения (дальности, времени, точностных характеристик, наличием различных типов информационных средств (ИС) для обнаружения, распознавания ИЭ, их целераспределения для обслуживания и т. п.) средств или способов противодействия (СП). Наличие большой номенклатуры ОЭ (а, соответственно, в их составе и СП) позволяет ОТС $\{B\}$ осуществлять оптимизацию стратегий своего поведения на основе распределения имеющегося ресурса ОЭ по возможным рубежам противодействия, исходя из пространственно-временных характеристик способов применения ИЭ [2]. При этом исходя из топологического построения подсистема противодействия ОТС $\{B\}$ в интересах обеспечения защиты группы объектов от ИЭ в ГД осуществляет в режиме централизованного управления распределение ресурса ОЭ (их СП) для обслуживания каждого ИЭ, а при защите одиночных объектов от ИЭ в ОД — в режиме децентрализованного (автономного) управления. Таким образом процесс преодоления ИЭ в ГД подсистемы противодействия ОТС $\{B\}$ представляется в виде совокупности последовательно или параллельно проводимых ОД с последующим обслуживанием каждого ИЭ назначенным типом ОЭ.

Основой для принятия решений в любой ОТС является информация — совокупность достоверных данных о составе, структуре, основных характеристиках и способах применения функционирующих в ее предметной области ОТС. Получение информации сопряжено с получением, обработкой и анализом данных от различного типа источников информации — информационных средств добывания информации и обеспечения управления ОЭ (ИЭ). Исходя из этого обеспечение устойчивого функционирования ОТС в конфликте может достигаться разрушением информации в иерархических контурах принятия решений ОТС $\{B\}$ за счет подавления наиболее важных

элементов и / или информационного воздействия на их ИС для снижения эффективности функционирования (качества информации) до некоторого минимального уровня. В этих условиях реализацию ОТС устойчивых действий целесообразно осуществлять, как показано в работе [1], на основе применения конфликтно-разрешающей обеспечивающей организационно-технической системы (ООТС). Цель ее применения состоит в прогнозе развития потенциально конкурирующих ОТС, обосновании стратегий поведения своей ОТС для различного уровня информационной определенности и парировании наступательных (оборонительных) действий конкурентов за счет информационного воздействия на их ИС для снижения качества информации. Основу ООТС составляет множество комплексов средств информационной безопасности (КСБ), которые, исходя из структуры решаемых задач, подразделяют на комплексы информационной индивидуальной (обеспечение действий отдельных ИЭ) и групповой (обеспечение действий групп ИЭ) безопасности элементов ОТС [1]. Их состав формируется на основе различного типа средств информационной безопасности (ИБ), например средств активных и пассивных помех, ложных целей, средств и способов дезинформации, имитации, тактических приемов, маневров и т. п. Эффект применения КСБ базируется на нарушении временного баланса функционирования подсистемы противодействия ОТС $\{B\}$, приводящего за счет снижения дальностей обнаружения, целераспределения и захвата (срыва) на сопровождение ИЭ ее информационных средств к изменению числа ОЭ, принимающих участие в отражении действий ИЭ, рубежей противодействия и вероятностей подавления ИЭ одним СП для заданных характеристик условий проведения одиночных или групповых действий.

Обоснование способов применения КСБ для обеспечения действий ИЭ ОТС в конфликте основывается на результатах оценки эффективности, получаемых с помощью соответствующих методов математического моделирования. Методы обоснования способов применения КСБ, исходя из возможности получения информации о характеристиках условий применения ОТС в конфликте, могут быть объединены в три группы.

1-я группа. Наиболее простыми являются детерминированные способы, совершенно не связанные с какими-либо случайностями, когда выбор некоторого закона применения КСБ на основе оценки результатов эффективности определяет условия устойчивого функционирования ОТС.

2-я группа. Более сложными являются способы применения КСБ, протекающие в условиях воз-

действия случайных факторов, законы распределения которых известны. В этом случае использование определенного закона применения КСБ по результатам оценки эффективности однозначно определяет распределение вероятностей устойчивых состояний ОТС.

3-я группа. Как правило, вероятностные законы распределения случайных факторов, процессы конфликтного применения ОТС являются уникальными, дискретными и конечными (заканчивающимися достижением одной из конкурирующих сторон минимального уровня жизнедеятельности) с последствием, а поэтому неизвестны. Одним из методов преодоления такого рода трудностей является обоснование способов применения КСБ в предположении, что распределение неизвестных факторов может быть определено из условия максимума или минимума некоторых показателей эффективности функционирования ОТС, т. е. задается не закон распределения неизвестных факторов, а принцип его определения. Наиболее целесообразным для разрешения конфликта ОТС является использование принципа максимина [7], определяющего поиск оптимальных способов применения КСБ для обеспечения функционирования ОТС на множестве варьируемых характеристик способов противодействия ОТС $\{B\}$: наличие централизованного, децентрализованного или автономного режимов управления для различного топологического построения подсистемы противодействия; эшелонированность по дальности (времени) применения ОЭ и ИС подсистемы противодействия; большое число типов ОЭ и перекрытие по дальности их зон эффективного применения, а также включение в структуру ОЭ (соответственно и их СП) ИС добывания информации и обеспечения управления в различных условиях конфликтной неопределенности. Наличие такого множества варьируемых характеристик позволяет ОТС $\{B\}$ осуществлять выбор оптимальных способов противодействия применению ИЭ ОТС и затрудняет проведение оценки эффективности и выбор целесообразных способов применения КСБ на основе различных вариантов состава средств информационной безопасности.

В настоящее время исследование способов применения КСБ для обеспечения действий ОТС в условиях конфликтной неопределенности в известных публикациях изучено недостаточно (в основном фрагментарно, исходя из предметного назначения ОТС) и рассматривается в такой математической форме, которая затрудняет использование полученных результатов для решения практических задач. Данные обстоятельства и предопределили научную актуальность и при-

кладную значимость предлагаемой статьи, направленной на разработку метода моделирования оценки эффективности применения КСБ. Результаты, полученные с помощью предлагаемого метода используются в качестве исходных данных в математической модели исследования эффективности применения ООТС при обеспечении действий ОТС различной целевой направленности в операциях [3].

В качестве интегрального показателя эффективности способов применения комплекса групповой безопасности (КГБ) при обеспечении защиты ИЭ в ГД ОТС $\{A\}$ в данном методе используется математическое ожидание числа ИЭ (U) из числа участвующих в групповых действиях (L), преодолевших противодействие ОТС $\{B\}$ и выполнивших поставленные задачи, а комплекса индивидуальной безопасности (КИБ) при обеспечении защиты ИЭ в ОД — вероятность преодоления одним ИЭ зоны действия ОЭ и выполнение поставленной задачи по воздействию на заданный объект применительно к характеристикам типового ядра конфликта уровня эпизода и ситуации, соответственно [4]. Данные показатели отражают основные особенности функционирования ОТС, являясь периодической функцией проводимых ОД (ГД), начального состояния и распределения в типовой ситуации (эпизоде) числа ИЭ, переводящих систему в каждом действии из начального в любое состояние. Эти показатели также являются чувствительными к применению разнообразных способов и средств ИБ в составе КСБ. При этом предполагается, что для каждого способа применения КСБ с k -м, $k = \overline{1, K}$, вариантом средств ИБ подсистемой противодействия ОТС $\{B\}$ выбираются стратегии поведения на основе оптимального распределения имеющегося разнородного ресурса ОЭ (их СП) на каждом и по всем возможным рубежам противодействия ИЭ в ГД и ОД в целевой и информационной обстановке условий типового ядра конфликта (эпизода или) ситуации. Эти стратегии минимизируют среднее (в смысле математического ожидания) значение показателя эффективности.

При разработке метода также предполагается, что используемые на каждом рубеже противодействия k -е варианты состава КСБ применяются оптимально с точки зрения максимизации эффективности преодоления ИЭ противодействия ОТС $\{B\}$.

В соответствии с этим постановка задачи состоит в определении математического ожидания ИЭ в ГД или ОД ОТС, преодолевших оптимально функционирующую на m -м, $m = \overline{1, M}$, направлении (районе) эшелонированную подсистему противодействия ОТС $\{B\}$ при применении k -х,

$k = \overline{1, K}$, вариантов состава КСБ и математически представляется в виде

$$U_N(k, m, B_N) = \max_{\{k\}} \min_{\{b_{Nij}\}} \sum_{i=1}^I P_{Nij}(k, m, b_{Nij}, d_{ijk}), \quad (1)$$

при ограничениях

$$\begin{aligned} \sum_{i=1}^I b_{Nij} &= B_{Nj}, \quad 0 \leq b_{Nij} \leq B_{Nj}, \\ d_{ijk}^{\min} &\leq d_{ijk} \leq d_{ijk}^{\max}, \quad n = \overline{1, N}. \end{aligned} \quad (2)$$

Здесь $U_N(k, m, B_N)$ — математическое ожидание (среднее число) ИЭ в ГД (ОД), преодолевших на m -м направлении действий n , $n = 1, \dots, N$, рубежей противодействия ОТС $\{B\}$ при k -м варианте состава КСБ (включает в свой состав, в том числе, и наиболее рациональные способы для преодоления зон действия наиболее "опасных" ОЭ по числу и эффективности обслуживаемых ИЭ) при оптимальном целераспределении ОТС $\{B\}$ ресурса B_N ОЭ (их СП) по всем ИЭ на n -м рубеже; d_{ijk} — вектор зоны противодействия j -го ОЭ при обслуживании i -го ИЭ в ОД (ГД) с k -м вариантом КСБ, $d_{ijk} = \|d_{ijk}^{\min}, d_{ijk}^{\max}\|_{IJK}$; $(d_{ijk}^{\min}, d_{ijk}^{\max})$ — зона противодействия по дальности j -го ОЭ при обслуживании i -го ИЭ в ОД (ГД) с k -м вариантом состава КИБ (КГБ); $P_{Ni}(k, m, b_{Nij}, d_{Nij})$ — вероятность преодоления i -м ИЭ с k -м вариантом КСБ в ОД или ГД ОТС на m -м направлении действий ОТС $\{B\}$ при назначении на него b_{Nij} ресурса СП j -го ОЭ, определяется в виде

$$P_{Nij}(k, m, b_{Nij}, d_{Nij}) = U_{N-1, ij}(k, m) \{1 - P_{on}(m) + P_{on}(m)[1 - Z_{mn} P_{nj}(m) K_{nij}]\}, \quad (3)$$

где Z_{mn} — параметр, учитывающий наличие n -го, $n = \overline{1, N}$, число рубежей противодействия на m -м, $m = \overline{1, M}$, направлении действий ИЭ в ОД или ГД ОТС,

$$Z_{mn} = \begin{cases} 1, & \text{если } n\text{-й рубеж находится на } m\text{-м} \\ & \text{направлении действий;} \\ 0, & \text{в противном случае;} \end{cases} \quad (4)$$

M — число направлений проведения ОД или ГД ОТС; $P_{on}(m)$ — вероятность обнаружения ИЭ в ОД или ГД информационными средствами ОТС $\{B\}$ на m -м направлении действий ОТС и n -м рубеже противодействия, $n = \overline{1, N}$, $m = \overline{1, M}$; $P_{nj}(m)$ — вероятность снижения эффективности применения ИЭ в ОД или ГД на m -м направлении действий ОТС и n -м рубеже противодействия одним СП j -го ОЭ; b_{nij} — число СП j -го ОЭ, используемых для обслуживания i -го ИЭ в ОД или ГД ОТС на n -м рубеже противодействия; K_{nij} — коэффи-

циент снижения вероятности применения i -го ИЭ на n -м рубеже противодействия j -м ОЭ при использовании k -го варианта l -го состава средств КСБ; $U_{N-1, ij}(k, m)$ — вероятность преодоления i -м ИЭ $(N-1)$ -го рубежа противодействия j -х ОЭ при использовании для его защиты k -го варианта КСБ на m -м направлении действий ОТС.

Сложность решения сформулированной задачи в виде (1) при ограничениях (2) состоит в том, что на каждом рубеже противодействия подлежит обслуживанию случайное число преодолевших предыдущий рубеж и обнаруженных ИС r -го числа ИЭ в ГД ОТС. Поэтому на каждом рубеже для определения математического ожидания числа преодолевших данный рубеж ИЭ, необходимо определять вероятность перехода $P_{пер.n}(\alpha_j, \alpha_s, B_{Nj})$ ИЭ в ГД из одного состояния α_s в другое α_j , где под состоянием ГД α_j (α_s) на n -м рубеже будем считать i -е число сохраненных из s ИЭ, пришедших на n -й рубеж в результате обслуживания s элементов на n -м рубеже B_{Nj} количества j -х ОЭ (их СП). Исходя из этого вероятность перехода ИЭ в ГД из состояния α_s в последующее α_j состояние вычисляется по формуле

$$\begin{aligned} P_{пер.m}(\alpha_j, \alpha_s, B_{Nij}) &= \sum_{r=s-j}^{s_0} P_n(\alpha_s) \times \\ &\times P_{обн.n}(\alpha_r, \alpha_s) P_{сохр.n}(\alpha_j, \alpha_r, \alpha_s, B_{Nij}), \quad (5) \\ s_0 &= \min\{B_{Nij}, s\}, \end{aligned}$$

где $P_n(\alpha_s)$ — вероятность прихода на n -й, $n = \overline{1, N}$, рубеж s -го количества ИЭ (равна средней вероятности нахождения ИЭ в ГД в состоянии α_s к n -му рубежу противодействия); $P_{обн.n}(\alpha_r, \alpha_s)$ — вероятность обнаружения ИС подсистемы противодействия ОТС $\{B\}$ на n -м рубеже r -го числа ИЭ в ГД из s (причем $P_{обн.n}(\alpha_r, \alpha_s) = 0$, если $r > s$), пришедших на n -й рубеж

$$P_{обн.n}(\alpha_r, \alpha_s) = C_s^r P_{on}^r (1 - P_{on})^{s-r}; \quad (6)$$

$$0 \leq r \leq s; \quad s = \overline{1, I};$$

C_s^r — число сочетаний из s по r ; $P_{сохр.n}(\alpha_j, \alpha_r, \alpha_s, B_{Nj})$ — вероятность сохранения j -го количества ИЭ в ГД к $(n+1)$ -му рубежу противодействия в результате обслуживания r ИЭ на n -м рубеже B_{Nj} разнородным ресурсом j -х ОЭ (их СП), при условии, что на n -й рубеж пришло s ИЭ

$$P_{сохр.n}(\alpha_j, \alpha_r, \alpha_s, B_{Nj}) = C_r^{j_0} P_{под.n}^{j_0} (1 - P_{под.n})^{r-j_0}; \quad (7)$$

$$0 \leq j_0 \leq r \leq s; \quad j = s - j_0; \quad j_0 = s - j;$$

$P_{\text{под.}n}$ — вероятность снижения эффективности применения ИЭ в ГД на n -м рубеже противодействия из r обнаруженных.

Исходя из этого вероятность прихода на $(n + 1)$ -й рубеж противодействия i -го ИЭ в ГД определяется в виде

$$P_{n+1}(\alpha_j) = \sum_{s=j}^I P_{\text{пер.}n}(\alpha_j, \alpha_s, B_{Nj}) = 1 - [1 - Z_{mn} P_n k_{nlij}]^{b_{ni}}, \quad (8)$$

где P_n — вероятность снижения эффективности применения ИЭ в условиях противодействия одного j -го ОЭ на n -м рубеже противодействия; b_{Nij} — число j -х ОЭ, назначенных для обслуживания одного ИЭ в ГД из r обнаруженных,

$$b_{Nij} = E[B_{Nj}/r] + 1 \text{ для } i = \overline{1, L}; \\ L = B_{Nj} - rE[B_{Nj}/r]; \\ b_{Nij} = E[B_{Nj}/r] \text{ для } i = \overline{L, r}; \quad (9)$$

$E[a]$ — целая часть числа a .

В приведенном (1)–(9) виде решение задачи осуществляется на основе теорий исследования операций [5], максимина [6], оптимального распределения ресурсов [8] и метода динамического программирования [7], обеспечивающих поиск оптимальных стратегий поведения эшелонированной подсистемы противодействия ОТС $\{B\}$ (при решении внутренней задачи минимизации в (1) по снижению эффективности применения ИЭ) и выборе максимального значения показателя эффективности (решение внешней задачи максимизации в (1)) при оптимизации способов применения k -х вариантов средств ИБ в составе КСБ.

Для упрощения расчетов в (8) и (9) возможно использование среднего значения ресурса оборонительных элементов (их СП), назначенных для обслуживания каждого ИЭ $b_{Nij} = E[B_{Nj}/r]$ для всех $i = \overline{1, r}$.

В результате последовательного пересчета числа исполнительных элементов, преодолевших в ГД каждый n -й рубеж противодействия до N -го включительно, по формуле (1) определяется значение интегрального показателя эффективности способов применения k -х вариантов КГБ при обеспечении преодоления ИЭ ОТС подсистемы противодействия ОТС $\{B\}$ для заданных характеристик средств ИБ и способов, используемых ИЭ на каждом рубеже и в целом на всех направлениях действий.

Исходными данными для проведения расчетов по оценке эффективности способов применения КСБ, являются:

- варианты состава и эффективность способов и средств ИБ по информационному воздействию на ИС добывания информации и обеспечения управлением ОЭ (их СП) на различных этапах обслуживания ИЭ;
- эффективность тактических приемов и маневров совершаемых ИЭ для преодоления зон действия ОЭ;
- типы, число и эффективность ОЭ по обслуживанию ИЭ на возможных рубежах противодействия;
- вероятность снижения эффективности подавления исполнительного элемента одним СП j -го ОЭ по отношению к эффективности способов и средств ИБ.

Получение этих данных основывается на результатах проведения структуризации и типизации условий конфликта на взаимообусловленную совокупность типовых ядер конфликта уровня эпизодов и ситуаций [4]. При этом исходные данные по характеристикам информационных средств и ОЭ (их СП) подсистемы противодействия ОТС $\{B\}$ принимаются на основе анализа и обобщения априорной информации, результатов прогноза и оценки с помощью простейших методов потенциальных возможностей ОЭ (их СП) по подавлению ИЭ ОТС данной предметной области.

Метод позволяет по интегральному показателю рассчитать эффективность выполнения различными типами ОТС поставленных задач, сформировать обоснованные предложения по рациональным вариантам состава средств КСБ, приемов и маневров и выбрать оптимальные способы применения КСБ.

Список литературы

1. Мистров Л. Е. Методический подход к моделированию действий функциональных организационно-технических систем в конфликте // Наука производству. 2004. № 11. С. 58–63.
2. Мистров Л. Е. Оценка эффективности иерархических обеспечивающих организационно-технических систем // Техника машиностроения. 2005. № 2. С. 40–58.
3. Мистров Л. Е. Математическая модель эффективности обеспечивающей организационно-технической системы // Приборы и системы. Управление, контроль, диагностика. 2007. № 1. С. 64–70.
4. Мистров Л. Е., Сербулов Ю. С. Методический подход к системному представлению многоцелевого конфликта функциональных иерархических систем // Информационно-измерительные и управляющие системы. 2005. № 3. Т. 3. С. 26–30.
5. Вентцель Е. С. Исследование операций. М.: Сов. радио, 1972.
6. Данскин Дж. Теория максимина и ее приложение к задаче распределения вооружения. М.: Сов. радио, 1970.
7. Беллман Р. Динамическое программирование. М.: Иностранная литература, 1960.
8. Берзин Е. А. Оптимальное распределение ресурсов и элементы синтеза систем. М.: Сов. радио, 1974. 303 с.



Л. Д. Агеев,

В. С. Калашников, д-р техн. наук, проф.,
Санкт-Петербургский университет
аэрокосмического приборостроения

Компьютерное проектирование проводящего рисунка плоских квазилогопериодических антенн

В классе сверхширокополосных антенн СВЧ значительное место занимают квазилогопериодические извилистые антенны, обладающие рядом преимуществ по сравнению со спиральными и логопериодическими антеннами.

Приведена методика компьютерного проектирования топологии проводящего рисунка этих антенн и подготовки данных, необходимых для их изготовления по любой планарной технологии. Методика включает в себя математическое обеспечение, позволяющее формализовать требования к проводящему рисунку антенны, и алгоритм расчета топологии этого рисунка.

Квазилогопериодические извилистые антенны имеют ряд преимуществ по сравнению со спиральными и логопериодическими антеннами (в частности, возможностью одновременной работы с электромагнитными волнами любых видов поляризации — вращающейся эллиптической левого и правого направлений вращения с коэффициентом эллиптичности, изменяющимся от нуля (линейная поляризация) до единицы (круговая поляризация)). Перспективность использования таких антенн в новейших телекоммуникационных системах отмечена в работе [1]. Наиболее полная информация о вариантах конструкций извилистых антенн содержится в работе [2]. Однако приведенные там сведения дают лишь общее представление об этих антеннах и не могут быть непосредственно использованы для разработки топологии проводящего рисунка извилистых антенн.

В настоящей работе приводится методика компьютерного проектирования топологии проводящего рисунка квазилогопериодических извилистых антенн и подготовки данных, необходимых для изготовления этих антенн по любой планарной технологии. Основу методики составляют математическое обеспечение, позволяющее формализовать требования к проводящему рисунку антенны, и алгоритм расчета топологии этого рисунка.

Описание математического обеспечения

Извилистые квазилогопериодические антенны, рассматриваемые в настоящей работе, представляют собой плоские структуры, вписанные в окружность радиуса R , проводящий рисунок которых состоит из четного числа идентичных плеч специального профиля, повернутых друг относительно друга вокруг общего центра на угол $360^\circ/K$, где K — число плеч.

Для знакомства с особенностями проводящего рисунка извилистой квазилогопериодической антенны выберем полярную систему координат (r, ϕ) и совместим ее центр с центром описанной окружности радиуса R . В основе проводящего рисунка каждого из идентичных плеч антенны лежит образующая ломаная (извилистая) кривая, ограниченная сектором, центральный угол которого равен 2Φ . Эта кривая состоит из N криволинейных элементов, соединенных концами и образующих друг с другом острые углы (рис. 1). Для обозначения элементов образующей ломаной кривой введем целочисленную переменную i , изменяющуюся в пределах от единицы до N с шагом 1; i -му элементу ломаной кривой соответствует текущий радиус-вектор r_i . Отсчет целочисленной переменной i начинается от элемента, имеющего самый большой радиус-вектор r_1 .

Плечо извилистой антенны получается в результате поворота образующей ломаной кривой на углы $\pm \delta$ относительно ее центральной оси и заполнения пространства между образовавшимися в результате этих поворотов границами пленочным проводником (рис. 2).

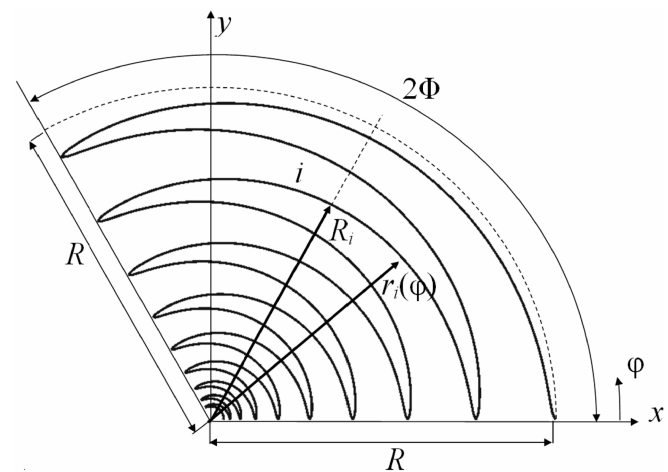


Рис. 1. Образующая ломаная кривая извилистой антенны ($2\Phi = 120^\circ$)

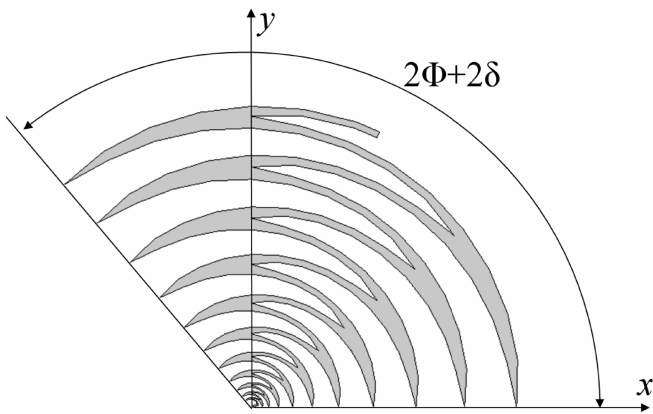


Рис. 2. Плечо извилистой антенны ($2\Phi = 90^\circ$, $2\delta = 40^\circ$)

Сама извилистая антенна, как было сказано выше, состоит из четного числа вышеупомянутых плеч, повернутых друг относительно друга вокруг общего центра на угол $360^\circ/K$ (рис. 3). Питание антенны осуществляется с помощью двухпроводных линий, концы которых присоединяются к плечам, развернутым друг относительно друга на угол 180° .

Образующие ломаные кривые извилистых антенн можно разделить на две группы:

- ломаные кривые, все элементы которых (от $i = 1$ до $i = N$) имеют одинаковую угловую ширину, равную 2Φ ;
- ломаные кривые, у которых угловая ширина элементов изменяется по различным законам $\Phi_i = f(i, \Phi)$, удовлетворяющим, однако, следующим условиям: $\Phi_1 = \Phi$, $\Phi_{i+1} < \Phi$.

Для криволинейных элементов образующих ломаных кривых, относящихся к первой группе, было разработано следующее математическое описание:

$$r_i = R_i \tau_i^{\{(-1)^{(i+1)}(1/\pi)\arcsin((\phi - \Phi)/\Phi)\}}, \quad (1)$$

где r_i — радиус-вектор текущей точки i -го элемента ломаной кривой (см. рис. 1); ϕ — полярный угол текущей точки i -го элемента ломаной кривой, изменяющийся в пределах от 0 до 2Φ (см. рис. 1); R_i — средний радиус-вектор i -го элемента ломаной кривой, соответствующий значению по-

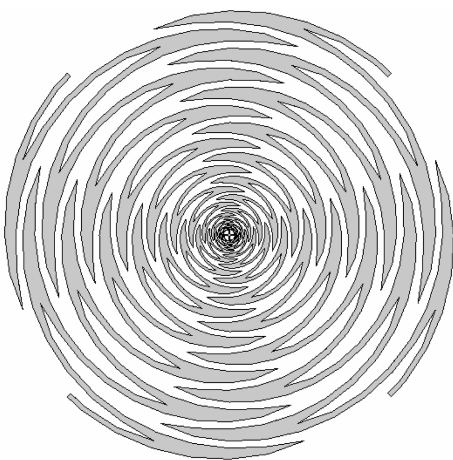


Рис. 3. Извилистая антенна ($K = 4$)

лярного угла $\phi = \Phi$ (см. рис. 1); τ_i — конструктивный параметр i -го элемента, определяющий "плотность" образующей ломаной кривой ($\tau_i < 1$).

Исходными данными для расчета образующей ломаной кривой являются следующие параметры: R — радиус окружности, в которую должен вписаться проводящий рисунок антенны; τ — конструктивный параметр образующей ломаной кривой, определяющий ее "плотность" ($\tau < 1$); N — число элементов образующей ломаной кривой; $\tau_i = f(i, \tau)$ — вид функции, определяющей зависимость τ_i от i и τ ; 2Φ — центральный угол сектора, который ограничивает образующую ломаную кривую.

Между параметрами первого элемента образующей ломаной кривой и исходными данными существует следующая связь:

$$\tau_1 = \tau, \quad \Phi_1 = \Phi, \quad R_1 = f(R, \tau). \quad (2)$$

Вид функции $R_1 = f(R, \tau)$ зависит от того, какая текущая точка первого элемента образующей ломаной кривой находится на окружности радиуса R . Если $r_1(\phi = 0) = R$, то $R_1 = R\sqrt{\tau}$ и образующая ломаная кривая имеет вид, изображенный на рис. 4, а. Если $r_1(\phi = \Phi) = R$, то $R_1 = R$ и образующая ломаная кривая имеет вид, изображенный на рис. 4, б. Если $r_1(\phi = 2\Phi) = R$, то $R_1 = R/\sqrt{\tau}$ и образующая ломаная кривая имеет вид, изображенный на рис. 4, в.

Независимо от вида формулы $R_i = f(R, \tau)$ средние радиус-векторы соседних элементов образующей ломаной кривой, начиная со второго ($i \geq 2$), связаны между собой следующим соотношением:

$$R_i = R_{i-1} \sqrt{\tau_i \tau_{i-1}}. \quad (3)$$

По терминологии, принятой в работе [2], антенны, у которых $\tau_i = \text{const} = \tau$, называются логопериодическими извилистыми антеннами, а антенны, у которых $\tau_i = f(\tau, i)$, — квазилогопериодическими извилистыми антеннами. В настоящей работе были рассмотрены следующие виды функций $\tau_i = f(\tau, i)$:

$$\tau_i = \tau^{a(i, m)}, \quad (4)$$

где $a(i, m) = i^m$, $m < 1$;

$$\tau_i = \tau S^{(i-1)}, \quad (5)$$

где $S < 1$;

$$\tau_i = \tau - \Delta\tau(i-1), \quad (6)$$

где $\Delta\tau(i-1) < \tau$;

$$\tau_i = (1 - i(1 - \tau))/(1 - (i-1)(1 - \tau)). \quad (7)$$

Функции вида (4)–(6) позволяют получить образующие ломаные кривые различной плотности. Функция (7) позволяет получить ломаную кривую, у которой разность между средними радиус-векторами соседних элементов постоянна ($R_i - R_{i+1} = \text{const}$).

Так как с использованием приведенного ниже программного обеспечения определение числен-

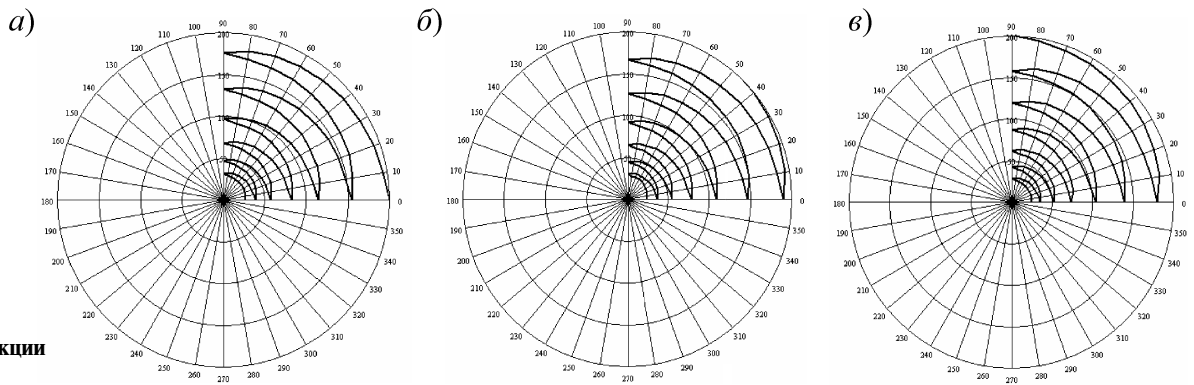


Рис. 4. Вид функции $R_1 = f(R, \tau)$

ных значений координат точек образующих ломаных кривых и их графическое построение осуществляется практически мгновенно, то, меняя исходные данные, можно быстро перебрать любое число вариантов построения проводящего рисунка и остановиться на том из них, который удовлетворяет всем требованиям разработчика.

При исследовании образующих ломаных кривых для функций (4)–(7) было обнаружено нарушение регулярности первых трех из них при малых значениях i , выразившееся в том, что разность w_i между R_i и R_{i+1} оказывалась меньше, чем разность w_{i+1} между R_{i+1} и R_{i+2} . Однако по мере роста i это нарушение исчезало. Поэтому в алгоритм расчета элементов ломаных кривых был введен блок проверки условия $w_i \geq w_{i+1}$, которое с учетом (3) трансформируется к следующему виду:

$$(1 - \sqrt{\tau_i \tau_{i+1}}) \geq \sqrt{\tau_i \tau_{i+1}} (1 - \sqrt{\tau_{i+1} \tau_{i+2}}). \quad (8)$$

Эта проверка позволила начинать строить образующую ломаную кривую с того значения i , для которого нарушение регулярности уже не имеет места. Для того чтобы образующая ломаная кривая достигала окружности радиуса R , рассчитывался нормировочный коэффициент $p = R/R_i$ (где i — определенное выше значение), на который умножался средний радиус-вектор i -го элемента и средние радиус-векторы всех элементов с большими номерами.

Математическое описание криволинейных элементов образующих ломаных кривых, относящихся ко второй группе (рис. 5), имеет следующие отличия от рассмотренного выше:

- центральный угол сектора, в который вписывается i -й криволинейный элемент, ограничен пределами $-\Phi_{i-1} \dots \Phi_i$;
- i -й криволинейный элемент состоит из двух частей, одна из которых лежит в области положительных значений угла ϕ (от 0 до Φ_i), а другая — в области отрицательных значений угла ϕ (от 0 до $-\Phi_{i-1}$), причем каждая из этих частей имеет свое математическое описание, различное для четных и нечетных значений параметра i .

Исходными данными для расчета образующей ломаной кривой являются следующие параметры:

R — радиус окружности, в которую должен вписаться проводящий рисунок антенны; τ — конструктивный параметр образующей ломаной кривой, определяющий ее "плотность" ($\tau < 1$); N — число элементов образующей ломаной кривой; Φ — центральный угол сектора, который ограничивает первую ветвь образующей ломаной кривой (для случая, когда $R_1 = R$, а $\Phi_1 = -\Phi$); $\tau_i = f(i, \tau)$ — вид функции, определяющей зависимость τ_i от i и τ ; $\Phi_i = f(i, \Phi)$ — вид функции, определяющей зависимость Φ_i от i и Φ .

Между параметрами первого элемента образующей ломаной кривой и исходными данными существует следующая связь:

$$\tau_1 = \tau, \Phi_1 = -\Phi, R_1 = R. \quad (9)$$

В этом случае формулы для расчета радиус-векторов текущих точек криволинейных элементов образующей ломаной кривой выглядят следующим образом.

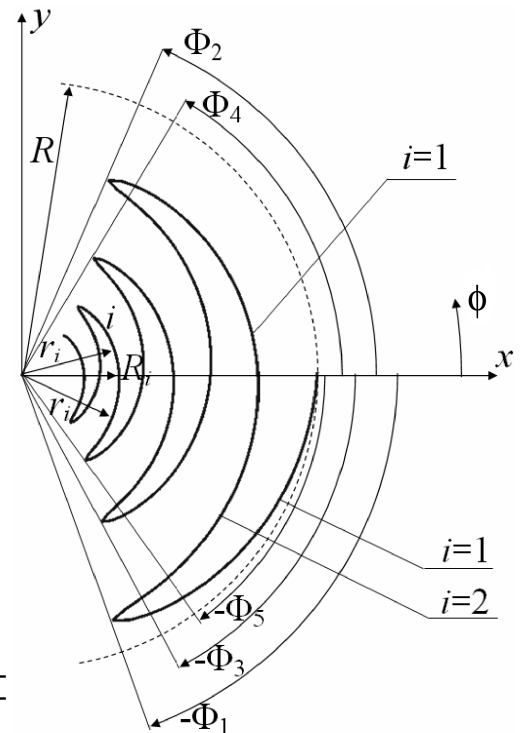


Рис. 5. Образующая ломаная кривая

Для четных i :

радиус-вектор текущей точки части криволинейного элемента, лежащей в области положительных значений угла ϕ (от 0 до Φ_i),

$$r_i = R_i \tau_i^{\{(1/\pi)\arcsin(\phi/\Phi_i)\}}; \quad (10)$$

радиус-вектор текущей точки части криволинейного элемента, лежащей в области отрицательных значений угла ϕ (от 0 до $-\Phi_{i-1}$),

$$r_i = R_i \tau_i^{\{(1/\pi)\arcsin(-\phi/\Phi_{i-1})\}}. \quad (11)$$

Для нечетных i :

радиус-вектор текущей точки части криволинейного элемента, лежащей в области положительных значений угла ϕ (от 0 до Φ_{i-1}),

$$r_i = R_i \tau_i^{\{(1/\pi)\arcsin(\phi/\Phi_{i-1})\}}; \quad (12)$$

радиус-вектор текущей точки части криволинейного элемента, лежащей в области отрицательных значений угла ϕ (от 0 до $-\Phi_i$),

$$r_i = R_i \tau_i^{\{(1/\pi)\arcsin(-\phi/\Phi_i)\}}. \quad (13)$$

Приведенные выше алгоритмы расчета координат образующих ломаных кривых были реализованы в программе MathCad. Пример образующей ломаной кривой первой группы, построенной в полярных координатах, приведен на рис. 1. Пример образующей ломаной кривой второй группы, построенной в полярных координатах, приведен на рис. 5.

Описание алгоритма расчета топологии проводящего рисунка антенны

Для получения топологии проводящего рисунка извилистой антенны необходимо выполнить следующие действия:

- провести расчет образующей ломаной кривой по приведенной выше методике в MathCad или в другой подобной программе, например, MatLab или Maple (результатом расчета являются массивы чисел, представляющие собой указанную образующую в декартовой системе координат);
- полученные массивы чисел необходимо сохранить в виде файла с расширением TXT, что можно реализовать при использовании программы Excel, входящей в состав пакета Microsoft Office;
- указанный TXT-файл необходимо импортировать в CAD-программу (Solid Works, Corel Draw...), и достроить проводящий рисунок извилистой антенны из образующей ломаной кривой, используя стандартный набор операций и функций программы.

Полученный проводящий рисунок может быть автоматически перенесен в существующие программы-моделировщики антенн (например, такие как CST Microwave Studio или Feko) и использован для построения компьютерной модели антенны. Затем средствами программы-моделировщика для этой модели могут быть рассчитаны ее радиотехнические характеристики.

Список литературы

1. Trends in Antennas for Wireless Communications // Microwave Journal. 2003. January P. 22–44. Special Report.
2. Dual polarized sinuous antennas: European Patent Application a. n. 36301175.5, priority 19.02.86 US 703043, p.n. 0198578 Al, Int. C1. H 01 Q 11/10.

УДК 004.627

Д. А. Плоткин, МГУ им М. В. Ломоносова, E-mail: darksun@mail.ru

Новый метод сжатия изображений, построенный на основе JPEG Baseline и метода бинарных интервальных преобразований

Предложен новый алгоритм сжатия статических изображений, в основе которого лежат JPEG Baseline и метод бинарных интервальных преобразований. Разработана программная реализация нового метода, предложены и реализованы несколько способов увеличения степени сжатия статических изображений. Основным достоинством предлагаемого метода сжатия является относительная вычислительная простота алгоритмов кодирования-декодирования: в алгоритмах используются не дорогостоящие операции умножения и деления, а только сложение, сравнение и сдвиг. Показано, что программная реализация нового метода сжатия статических изображений превосходит JPEG Baseline на многих типах изображений.

Введение

При решении задач спутниковой телеметрии, компьютерной обработки рентгеновских изображений, в мобильной телефонии, в цифровом те-

левидении и многих других постоянно возникает необходимость использовать сжатие данных. Разработка и исследование новых методов сжатия информации является актуальной научной и прикладной задачей, обусловленной развитием и со-

вершенствованием вычислительной техники. Алгоритмы компрессии должны в полном объеме использовать все средства, предоставляемые современными компьютерами, но в то же время должна быть обеспечена совместимость и с более простой аппаратурой, находящейся в эксплуатации. И конечно же, эти алгоритмы не должны уступать по эффективности широко известным алгоритмам компрессии, работающим на простой аппаратуре.

Метод бинарных интервальных преобразований и основанное на нем универсальное кодирование

Метод бинарных интервальных преобразований был подробно исследован при сжатии бинарных файлов. Были найдены оптимальные параметры сжатия метода, позволяющие добиться наилучших результатов, что было подробно изложено в статье [1]. Целью данной публикации является развитие работ, показанных в статье [1], по сжатию бинарных файлов. Построен новый алгоритм сжатия статических изображений, в основе которого лежат JPEG Baseline и метод бинарных интервальных преобразований.

В основе универсального метода сжатия файлов лежит алгоритм бинарных интервальных преобразований [2, 3], имеющий малую алгоритмическую сложность и показывающий хорошие результаты сжатия. Для достижения наилучших результатов сжатия необходимо подобрать параметры метода оптимальным образом. В связи с новизной метода не существует теоретических результатов, позволяющих определить эти оптимальные параметры, при которых достигается максимальное сжатие. Сложность заключается в многомерности задачи и отсутствии простых зависимостей между параметрами. Поэтому поиск данных оптимальных параметров осуществлялся экспериментально. У метода бинарных интервальных преобразований два главных параметра сжатия — мощность алфавита и порядок следования "обобщенных" букв. Но сами эти параметры могут варьироваться в большом диапазоне значений. В результате исследования были выработаны рекомендации по поиску оптимальных параметров для различных типов файлов.

- для бинарных файлов надо последними кодировать "нулевые" буквы;
- для текстовых файлов "нулевые" и "единичные" буквы должны кодироваться первыми, но никак не последними.

Использование неоптимальных параметров приводит к резкому увеличению размера файла, поэтому нахождение их правильных значений имеет очень большое значение.

В основе универсального метода сжатия данных лежит бинарное интервальное преобразование, метод стопки книг и преобразование Барроуза—Виллера [4]. Последние два метода не сжимают данные, однако осуществляемые ими преобразования приводят к улучшению сжатия данных последующим применяющимся статическим кодером.

Было проведено сравнение метода бинарных интервальных преобразований с арифметическим кодированием и кодированием по Хаффману, а также универсального метода с известными алгоритмами zip и bzip2. Результаты сжатия [5], полученные с помощью бинарного интервального преобразования с оптимальным подбором параметров, занимают промежуточное положение между кодированием по Хаффману и арифметическим кодированием, имеющим большую арифметическую сложность по сравнению с бинарными интервальными преобразованиями. Универсальное кодирование показало результаты сжатия лучшие, чем алгоритмы zip и bzip2. Сравнение проводилось на стандартном наборе Calgary Corpus [6]. Универсальность кодирования заключается в одинаково хороших результатах сжатия вне зависимости от типа сжимаемого файла.

Применение метода бинарных интервальных преобразований при сжатии статических изображений

Подробно исследованный на бинарных файлах метод бинарных интервальных преобразований, а также полученные результаты, превосходящие результаты широко известных методов, позволили начать исследование статических изображений — другой не менее важной области мультимедийных данных. Новый метод M-JPEG основан на JPEG Baseline [7] и бинарных интервальных преобразованиях.

В новом методе M-JPEG по сравнению с JPEG Baseline изменились только модули, отвечающие за статическое кодирование — кодирование по Хаффману. На первых шагах работы алгоритма JPEG исходное изображение уже разбито на макроблоки размером 16×16 пикселей. Данные, находящиеся в этих матрицах, считывались "зигзаг" сканированием и направлялись на вход энкодера бинарных интервальных преобразований. В связи с тем, что бинарное интервальное преобразование является статическим методом кодирования, и сжатие достигается только при некотором объеме исходной информации, то все полученные "зигзаг" сканированием одномерные вектора записывались во входной поток кодера.

После записи информации из всех макроблоков начинается кодирование в соответствии с за-

ранее установленными параметрами сжатия, определенными в ходе проведенного анализа, описанного в статье. Кодирование проводится для алфавита мощности, равной четырем. Порядок следования "обобщенных" букв был установлен тот, который оказался наилучшим для бинарных файлов, так как именно эта категория файлов наиболее точно отражает содержание информации, поступающей на вход энкодера.

Для улучшения характеристик сжатия M-JPEG был предложен новый способ передачи информации в поток. После считывания данных из матрицы они делятся на две последовательности по принципу ноль/не ноль. В первой последовательности единичный бит ставится вместо ненулевого числа, а нулевой бит — вместо нуля. Во вторую же последовательность записываются все ненулевые числа. Первая последовательность отправляется на вход энкодера программы, а вторая кодируется по Хаффману. Например, из матрицы было считано: "12, 0, 3, 4, 0, 0, 0, 0, 5, 0, 8, 0, 0, 1, 0". Тогда будет сформировано две последовательности следующего вида:

- первая: "101100001010010" — итого 15 бит.
- вторая: "12, 3, 4, 5, 8, 1" — 6 чисел.

Благодаря предварительной обработке информации в последовательности, идущей на вход программы сжатия, получается большее число нулей. Данные становятся более подходящей структуры для обработки с помощью бинарных интервальных преобразований.

Такой подход позволил улучшить показатели сжатия статических изображений (см. рисунок на третьей стороне обложки) алгоритмом M_JPEG до 15 % (см. на рисунке — M_JPEG(v1)).

Следующим шагом в совершенствовании алгоритма M-JPEG стал подбор таблиц AC и DC [7] в зависимости от информации, обрабатываемой кодером. Таблицы AC и DC стали использоваться для коэффициентов AC и DC соответственно. Помимо этого был предложен новый способ получения собственных таблиц для коэффициентов AC и DC. Значения элементов макроблоков не могут быть по модулю больше 2048. Это связано с особенностями дискретного косинусного преобразования, применяемого на ранних стадиях JPEG-кодирования. Поэтому вся область значений от -2048 до 2048 была разбита на 12 интервалов, для которых считались частоты появления. Таблицы для коэффициентов AC и DC [8, 9] строились с учетом статистики частоты появления данных в интервалах таким образом, чтобы данные, частота появления которых наибольшая, кодировались словами наименьшей длины.

Такой способ построения собственных таблиц позволил улучшить показатели степени сжатия

изображений еще до 10 % (на рис. — M_JPEG(v2) и M_JPEG(v3)).

Был предложен еще один способ для улучшения сжатия M-JPEG — удаление избыточности из закодированной информации. После дискретного косинусного преобразования и квантования нижняя половина макроблока часто бывает полностью обнулена. При считывании таких матриц "зигзаг" сканированием образуется большое число нулей в конце одномерного вектора. Записывать и кодировать все эти нули излишне — достаточно поставить специальный символ, означающий конец макроблока. Этот символ должен быть уникальным и не встречаться больше в последовательности. Поскольку значения элементов макроблока не превосходят по модулю 2048, то таким символом может являться любое число большее 2048. При декодировании алгоритм, на вход которого поступает такой символ, понимает, что все оставшиеся неопределенными значения макроблока необходимо заполнить нулями. Такой способ удаления избыточности позволил еще до 15 % (на рис. — M-JPEG(v3.1)) лучше сжимать статические изображения по сравнению с JPEG Baseline.

В результате проведенных исследований на тестовом наборе изображений Waterloo Repertoire [6] были показаны результаты, превосходящие JPEG Baseline на многих типах изображений. На рисунке показана динамика изменений степени сжатия M-JPEG после применения соответствующих улучшений алгоритма.

Список литературы

1. **Плоткин Д. А.** Оптимизация параметров в методе бинарных интервальных преобразований // Информационные технологии. № 11. 2006. С. 66–71.
2. **Брайловский И. В.** Эффективное сжатие данных с помощью метода обобщенных интервальных преобразований / Дисс. на соискание ученой степени канд. ф.-м. наук. М.: ИМВ РАН, 2003.
3. **Braïlovsky I., Kravtsov E., Plotkin D.** A new low complexity entropy coding method // Proc. of 14th International Conference of Computer Graphics and Vision. Moscow State University, 2004.
4. **Burrows M., Wheeler D. J.** A Block-sorting Lossless Data Compression Algorithm // Digital System Research Center, Research report 124. 1994. P. 1–25.
5. **Плоткин Д. А.** Эффективное сжатие данных с помощью бинарного интервального кодирования // Сборник тезисов лучших дипломных работ 2005 г. факультета ВМиК МГУ им. М. В. Ломоносова. М.: Изд. ВМиК МГУ, 2005. С. 81–82.
6. **Calgary Corpus, Waterloo Repertoire** — <http://www.data-compression.info/>
7. **Rao K. R., Hwang J. J.** Techniques and Standards for Image, Video, and Audio Coding. Prentice Hall PTR, 1996. P. 1–27.
8. **Shannon C.** A mathematical theory of communication // Bell System Technical Journal. 1948. V. 27. P. 379–423.
9. **Хаффман Д. А.** Метод построения кодов с минимальной избыточностью: Пер. с англ. // Кибернетический сборник. Вып. 3. 1961. С. 50–57.



УДК [004.4'24:519.87]::004.272.2

В. В. Пекунов, канд. техн. наук,
Ивановский государственный
энергетический университет,
Региональный НОЦ по наноматериалам
"Жидкие кристаллы" (г. Иваново)

Автоматизация параллельного программирования при моделировании многофазных сред

Предлагается гибкий подход к автоматизации параллельного программирования при решении задач механики многофазных сплошных сред. Подход основан на применении объектно-событийной модели порождения программ. Описывается структура модели. Формулируются принципы интерпретации и технологии трансляции моделей в программный код.

Введение

В последние десятилетия программное обеспечение активно развивается, что обусловлено расширением сферы применения компьютерной техники. Увеличение потребности в качественном программном обеспечении побуждает искать пути полной или частичной автоматизации его разработки и модификации. Известен ряд работ в области автоматизации программирования, предлагающих различные пути решения данной проблемы. Сюда относятся:

- модульный подход с планированием в пространстве состояний, позволяющим определить последовательность вызова необходимых модулей (система ПРИЗ [1] с построением описания задачи на языке УТОПИСТ);
- модульный подход, основанный на анализе постановки задачи на естественном языке с последующей сборкой кода из готовых блоков (система IPGS [2, 3]);
- автоматный подход, основанный на исчерпывающем перечислении возможных состояний программы (разработки на базе SWITCH-технологий [4]);
- объектный подход с прямым исполнением, согласно которому программа представляется в виде диаграммы специализированных объек-

тов, непосредственно исполняемых на виртуальной машине (технология Флора/FloraWare, разработка компании "Компас Плюс"). Также можно упомянуть системы (например Rational Rose), позволяющие описывать структуру и логику программы посредством UML-диаграмм и генерировать на их основе "скелет" кода, дополняемый программистом.

Данная работа посвящена проблеме автоматизации программирования при решении задач параллельного численного моделирования динамики многофазных сплошных сред (аэрогидродинамики и теплообмена [5]). Характерными особенностями таких задач являются: высокая степень формализации; существенная трудоемкость вычислений и, как следствие, высокие требования к эффективности параллельного численного решения. Первое свойство предполагает простоту и естественность постановки исходной задачи в формальном виде, что исключает необходимость в выполнении предварительных сложных процедур формализации, таких как анализ текстов на естественном языке [2, 3]. Прочие свойства говорят о необходимости гибкого подхода к порождению параллельной программы, требующего:

- выбора наиболее оптимальных стратегий распараллеливания;
- сложного комбинирования и настройки алгоритмов, реализующих различные методики численного интегрирования [5].

Указанные выше основные подходы к автоматизации программирования не отвечают поставленным требованиям. Модульные подходы не являются достаточно гибкими, поскольку не дают возможности осуществить сложную настройку алгоритмов, которую затруднительно реализовать путем простого изменения параметров вызова соответствующих типовых процедур или комбинаторного перечисления возможных вариантов таких процедур. Автоматный подход с перечислением состояний в наибольшей степени пригоден для решения задач управления (в том числе при разработке интерфейсов программных систем), но не для трудоемких вычислительных задач. Объектный подход с прямым исполнением неэффективен для наших задач, поскольку предполагает следующие дополнительные временные затраты: на работу виртуальной машины; на взаимодействие объектов (скорее всего распараллеливание будет

проводиться динамически, исходя из идеи портфеля задач [6]).

Актуальна задача разработки гибкого подхода к автоматизации программирования, позволяющего эффективно решать задачи параллельного численного моделирования динамики многофазных сплошных сред в соответствии с указанными требованиями.

1. Общий подход к автоматизации программирования

Пусть предметная область представлена в виде набора базовых понятий различной степени общности. Согласно определению понятие — мысль, обобщает и выделяет предметы некоторого класса по определенным общим признакам. Следовательно, предметная область может быть описана как иерархия классов (понятий), описывающих общие свойства объектов, подпадающих под данное понятие. Тогда формальным описанием исходной задачи будет совокупность взаимосвязанных объектов, являющихся экземплярами классов соответствующей предметной области.

Такое описание целесообразно представить в виде блочной схемы, легкой для разработки и модификации. Схема может служить основой для генерации программы, если считать, что каждый входящий в нее объект порождает некий программный ресурс (функцию, переменную, цикл и т. п.), представленный одним или несколькими фрагментами программы, которые могут перемежаться с другими фрагментами. Содержание порождаемых фрагментов в общем случае будет определяться структурой связей между объектами и их параметрами. Порядок компоновки фрагментов определим не только структурой связей, но и интерпретацией описания как событийной модели, в которой каждое событие определяет некий этап генерации кода.

Данный подход обеспечивает достаточную гибкость порождения программ. Анализируя структуру модели, объекты могут выбрать оптимальные стратегии распараллеливания, скомбинировать и настроить стандартные алгоритмы численного интегрирования, осуществить контекстную генерацию вспомогательного кода.

2. Структура модели порождения программ

Предлагается объектно-событийная модель, используемая как для описания задачи, так и для порождения программы. Модель представляется графом (P, E) , где P — множество узлов (объектов различных классов); E — множество дуг (связей между объектами). На множестве P определим функцию принадлежности к одному из классов: $\text{class}: P \rightarrow C$, где C — множество классов.

Существуют два основных типа связей между объектами: основные и вспомогательные. Основные связи определяют порядок срабатывания объектов и являются "каналами" передачи данных. Вспомогательные связи — лишь констатация факта наличия связи, отражающей структурные аспекты модели. Связи имеют различную семантику: "используется для", "входит в", "формирует", "имеет отображение" и другие. Любая связь исходит из выходного контакта объекта и входит во входной контакт объекта.

Граф модели может содержать циклы, но каждый цикл должен содержать хотя бы одну вспомогательную связь. Подграф, построенный лишь на основных связях, представляет собой сеть — циклы по основным связям не допускаются.

Класс S представляет собой (рис. 1) пятерку (N_c, I, O, F, M) , где N_c — идентификатор (имя) класса; I и O — наборы входных и выходных контактов (определим функциями $I = \text{in}(S)$, $O = \text{out}(S)$); F — поля; M — порождающие методы. Определим отношение иерархии $\text{parent}: C \rightarrow C$, заданное функцией

$$\forall a \in C \exists \text{parent}(a) = \begin{cases} (\varepsilon, \emptyset, \emptyset, \emptyset, \emptyset), & \text{если } a \text{ есть корень иерархии;} \\ c \in C, & \text{если } c \text{ есть предок } a, \end{cases}$$

где ε — пустая цепочка.

Контакты, поля и методы любого класса могут быть собственными, унаследованными или переопределенными. То есть, если $S = (N_c^S, I^S, O^S, F^S, M^S)$; $Q = (N_c^Q, I^Q, O^Q, F^Q, M^Q)$, причем $Q = \text{parent}(S)$, то $I^Q \subseteq I^S$, $O^Q \subseteq O^S$, $F^Q \subseteq F^S$, $M^Q \subseteq M^S$. В соответствии с принципами объектного подхода возможен вызов методов-предков из M^Q в соответствующих переопределенных методах M^S .

Любой контакт $Z \in \text{in}(S) \cup \text{out}(S)$ класса S является шестеркой $(N_p, T, L, \text{Min}, \text{Max}, D)$, где N_p —

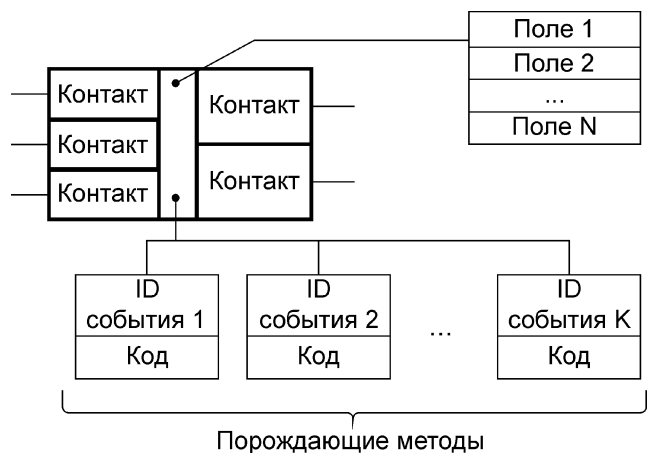


Рис. 1. Схематическое изображение класса

идентификатор контакта; $T \in \{\text{входной, выходной}\}$ — тип контакта; L — множество допустимых выходных связей: $L = \{(s, z) | Z \in \text{out}(S) \ \& \ s \in C \ \& \ z = \text{in}(s) \ \& \ [\text{допустима связь } (S, Z) \rightarrow (s, z)]\}$; $\text{Min} \in \{0, 1\}$ — минимальная степень контакта; $\text{Max} \in \{1, \infty\}$ — максимальная степень; $D = D[N_k]$ — ассоциативный кортеж (буфер передачи данных), отражающий состояние контакта объекта при порождении программы, причем N_k — имя k -й ячейки кортежа.

Множество полей F класса S можно также представить в виде ассоциативного кортежа $F = F[N_j]$, хранящего текущее состояние объекта, принадлежащего классу S , причем N_j — имя j -й ячейки кортежа.

В множестве M класса S каждый метод является программным скриптом, реализующим реакцию на некоторое событие, произошедшее в процессе интерпретации модели. Удобно представить данное множество в виде вектора $M = (M_{S1}, M_{S2}, \dots, M_{SN})$, где N — размер календаря событий. При реализации целесообразно объединить все методы в один скрипт, определив реакцию на каждое i -е событие с помощью конструкции выбора. Реакция на событие может включать действия трех типов, первые два из которых являются необязательными:

- 1) контекстная генерация фрагмента кода на основании значений полей F и информации, поступившей на его входные контакты и хранящейся в соответствующих кортежах D . Если данное действие не выполняется, то считается, что порожден пустой фрагмент;
- 2) планирование нового события (включение его в календарь событий);
- 3) изменение значений в ассоциативных кортежах D выходных контактов.

Отметим, что для удобства представления модели в виде блочной схемы существует специальный класс-контейнер $S^* = (N_c^*, I^*, O^*, \emptyset, \emptyset)$, представляющий "обертку" для произвольного разработанного ранее фрагмента модели (P^*, E^*).

При этом $I^* \subseteq \{t | t \in \text{in}(s) \ \& \ s \in Q^*\}$;
 $O^* \subseteq \{t | t \in \text{out}(s) \ \& \ s \in Q^*\}$;
 $Q^* = \{q | q = \text{class}(p) \ \& \ p \in P^*\}$.

При интерпретации модели контейнеры непосредственно заменяются фрагментами (P^*, E^*), включаемыми в общую модель (P, E).

3. Интерпретация модели

Интерпретация модели осуществляется на двух взаимосвязанных уровнях, которые условно назовем событийным и объектно-сетевым. Схематически процесс интерпретации показан на рис. 2, где окружностями с цифрами обозначены объекты модели, пустыми квадратами — методы, штрихованными

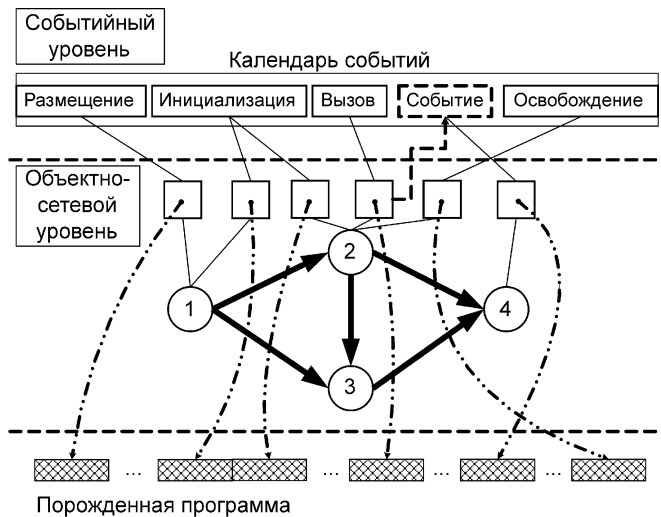


Рис. 2. Схема процесса интерпретации модели и порождения программы

прямоугольниками — фрагменты программы, порожденные соответствующими методами.

На событийном уровне процесс интерпретации управляется календарем событий, представленным вектором $C = (C_1, C_2, \dots, C_N)$. В самом начале интерпретации календарь содержит четыре обязательных события, соответствующих основным этапам работы с любым ресурсом: размещение, инициализация, вызов, деинициализация. Событие "размещение" всегда происходит первым (автоматически). Произошедшее событие интерпретируется на объектно-сетевом уровне, после чего происходит следующее событие. Интерпретация модели заканчивается, если исчерпан календарь событий.

В процессе интерпретации i -го события (объектно-сетевом уровне) используется методика, подобная применяемой для сетевых графиков работ. Последовательно срабатывают группы объектов модели. В первую очередь срабатывают объекты, не имеющие входных контактов ($I = \emptyset$), а также объекты, к входным контактам которых не подсоединены основные связи. При срабатывании j -го объекта активизируется его метод $(M_j)_i$, ответственный за обработку i -го события, который генерирует фрагмент программы (символьную цепочку) $(Z_j)_i$, формирует данные на своих выходах и, возможно, планирует новые события. Все прочие объекты срабатывают тогда и только тогда, когда сработают все объекты, связанные по основным связям с их входными контактами.

При выполнении порождающих методов объект может поместить произвольный массив значений в любую ячейку кортежей $D[N_k]$ своих выходных контактов и прочитать массив значений из любой ячейки $D[N_k]$ входных контактов. Чтобы объекты имели информацию о структуре связей модели, система автоматически помещает в

специальные ячейки выходных кортежей следующие данные: идентификатор контакта N_p , идентификатор объекта, которому принадлежит контакт, и идентификатор класса N_c объекта.

Рассмотрим правила передачи данных по связям. Если ко входному контакту $A = (N_t^A, T^A, L^A, \text{Min}^A, \text{Max}^A, D^A)$ одного объекта подключены выходные контакты $B^i = (N_t^i, T^i, L^i, \text{Min}^i, \text{Max}^i, D^i)$, $i = \overline{1, m}$, иных объектов, то в ячейку кортеж $D^A[N_k]$ контакта A помещается копия результата слияния ячеек кортежей $D^i[N_k]$ контактов B^i :

$$\forall N_k \in \bigcup_{i=1}^m V^i: D^A[N_k] = D^1[N_k] \circ D^2[N_k] \circ \dots \circ D^m[N_k],$$

где V^i — множество имен ячеек кортежей D^i ; знаком "o" обозначена операция слияния массивов.

Интерпретация события заканчивается, когда сработают все объекты модели. Последовательность срабатываний объектов может быть описана вектором Q , элемент q_k которого содержит номер объекта, сработавшего k -м по счету. Если несколько объектов сработали одновременно, то их номера располагаются в векторе Q подряд и их последовательность не имеет значения.

По окончании интерпретации i -го события формируется символьная цепочка

$$Y_i = (Z_{q_1})_i + (Z_{q_2})_i + \dots + (Z_{q_n})_i$$

где n — число объектов модели, знаком "+" обозначена операция слияния цепочек.

Порожденная программа G определяется символьной цепочкой

$$G = Y_1 + Y_2 + \dots + Y_N,$$

где N — общее число произошедших событий.

Описанные модель и схема ее интерпретации позволяют избежать существенных неоднозначностей порядка компоновки символьных цепочек, связанных с циклическим срабатыванием элементов модели (например, при использовании кусочно-линейных агрегативных моделей). Поддерживается возможность неоднократного срабатывания объектов за счет ввода в модель событийной составляющей.

Объектно-событийная модель позволяет породить любую программу, если ее алгоритм может быть представлен в виде графа указанного выше вида (P, E) .

4. Трансляция модели

Сначала всегда проводится интерпретация модели, результатом которой является программа на конкретном (например, Pascal, C, язык

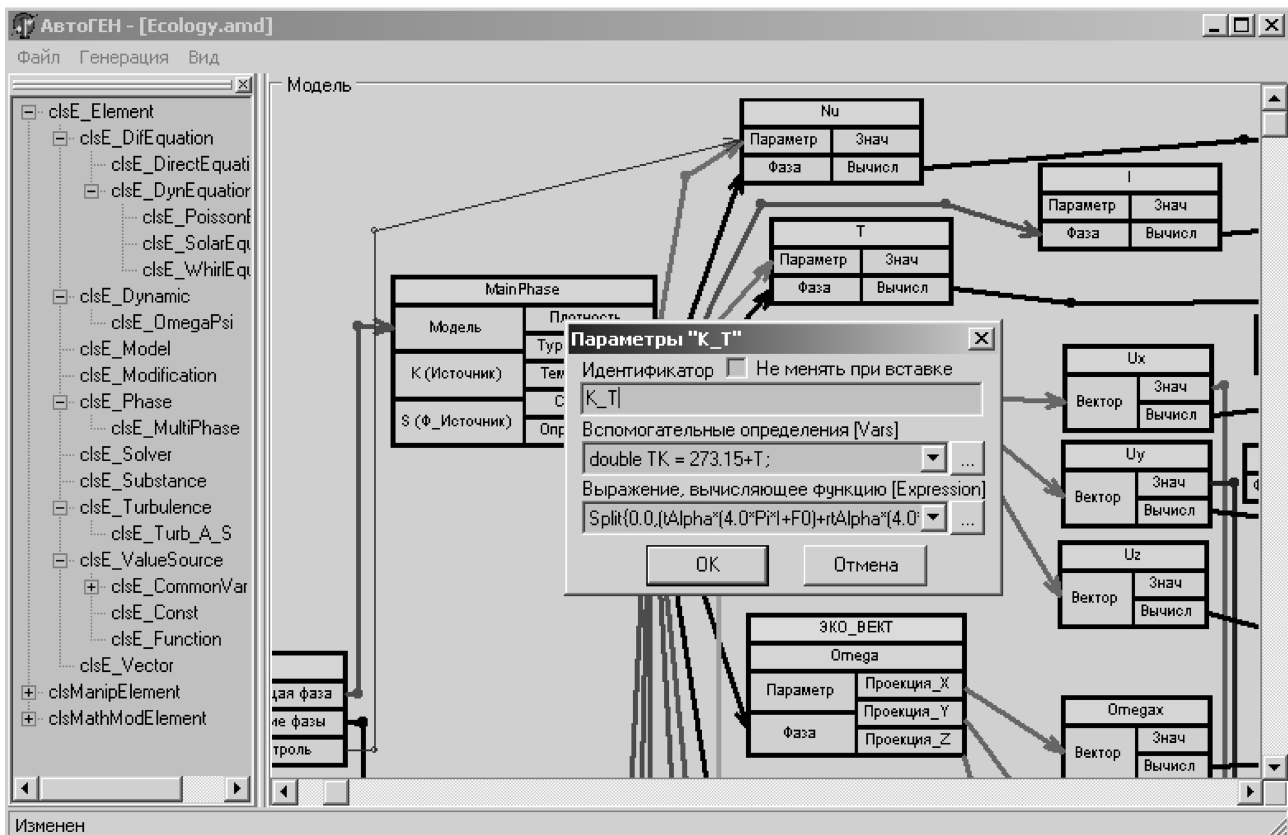


Рис. 3. Фрагмент модели в окне системы автоматизированного порождения программ

М-скриптов системы MatLab) или на специальном обобщенном алгоритмическом языке. В первом случае получаем готовую к исполнению программу, во втором случае необходима дополнительная трансляция обобщенной программы на конкретный язык программирования. При всей простоте первого подхода нельзя не отметить его серьезный недостаток: порождающие методы ориентируются на конкретный язык. Если возникает потребность в генерации программы на ином языке, то приходится вносить изменения в реализацию методов, что требует существенных временных затрат. Второй подход является более гибким (специфические конструкции языка генерируются транслятором), но более сложным, поскольку возможности обобщенного языка существенно ограничены именно в силу его общности. Однако при этом разработка порождающих методов проводится однократно, что уменьшает временные затраты.

5. Иерархия классов для задач параллельного численного моделирования многофазных сред

Разработана иерархия классов объектно-событийной модели, позволяющая создавать многофазные многокомпонентные модели (в частных производных) процессов распространения загрязнений в воздушной среде, на базе которых генерируются программные модули, подключаемые к соответствующему параллельному решателю. В классы модели интегрированы стратегии, позволяющие выбрать наиболее эффективный метод распараллеливания [5] и оптимизировать схему обменов данными при распараллеливании по пространству.

Заключение

В данной работе предложен гибкий подход к автоматизации программирования, позволяющий эффективно решать задачи параллельного численного моделирования динамики многофазных сплошных сред. Предложена объектно-событийная модель порождения программ, отличающаяся гибкой схемой контекстной компоновки фрагментов порождаемой программы. Описан алгоритм интерпретации и технологии трансляции модели. Данный подход позволяет осуществлять сложную настройку комбинируемых алгоритмов численного интегрирования и успешно приме-



Рис. 4. Пример результата трансляции модели на язык C

нять достаточно совершенные стратегии, обеспечивающие оптимальное распараллеливание.

В настоящее время разработана первая версия системы автоматизированного порождения программ (рис. 3, 4), базирующейся на описанном в данной работе подходе. Система включает: графический редактор, обеспечивающий визуальное проектирование модели; интерпретатор моделей; транслятор с обобщенного алгоритмического языка на язык C или Pascal.

В качестве языка порождающих методов использован скриптовый язык RHP, что обусловлено следующими факторами: а) данный язык ориентирован на порождение текстовых фрагментов; б) это полноценный алгоритмический язык, реализующий процедурный и объектный подходы; в) существуют свободно распространяемые интерпретаторы данного языка. Транслятор обобщенной программы написан на языке SNOBOL, ориентированном на обработку строк, лексико-синтаксический анализ и компиляцию.

С использованием данной системы была разработана программа параллельного численного моделирования образования и распространения твердых, жидких и газообразных загрязнителей AirEcology-P¹. Предложенный нами подход позволил упростить оперативную модификацию рас-

¹ Зарегистрирована в РОСПАТЕНТ, свидетельство № 2006611068, 21.03.2006.

четной модели под конкретные условия эксперимента (ввод и удаление фаз, уравнений, подмоделей). В частности, в двухфазную модель распространения загрязнений, описанную в работе [5], были включены третья капельная фаза и подмодель переноса излучения, затем была порождена соответствующая параллельная программа моделирования.

Работа выполнена при финансовой поддержке Минобразования и науки (грант РНП.2.2.1.1.7280).

Список литературы

1. **Кахро М. И., Каля А. П., Тыгу Э. Х.** Инструментальная система программирования ЕС ЭВМ (ПРИЗ). М.: Финансы и статистика, 1988.

2. **Зубков В. П., Назаретский С. П.** IPGS — интеллектуальная система автоматизированного программирования // Информационная среда вуза: Сб. статей. Иваново: ИГАСА, 2000. С. 213—215.

3. **Аленкин А. А., Зубков В. П.** Как автоматизировать процесс программирования // Мир ПК. 2002. № 8. С. 126—129.

4. **Шальго А. А., Туккель Н. И.** Программирование с явным выделением состояний // Мир ПК. 2001. № 8. С. 116—121; № 9. С. 132—138.

5. **Пекунов В. В., Ясинский Ф. Н.** Параллельное решение задачи численного моделирования распространения загрязнений в воздушном бассейне большого города и в окрестности предприятия. Препринт. № 36. М.: Инст. прикл. матем. им. М. В. Келдыша РАН, 2003.

6. **Эндрюс Г. Р.** Основы многопоточного, параллельного и распределенного программирования. М.: Изд. дом "Вильямс", 2003.

УДК 004.272

Д. И. Черемисинов, канд. техн. наук,
Объединенный институт проблем информатики,
Минск, Беларусь
E-mail: cher@newman.bas-net.by

Разработка параллельных программ для решения логико-комбинаторных задач

Обсуждаются методы отладки и предотвращения ошибок при разработке параллельных программ для кластерного компьютера и реализующие алгоритмы решения логико-комбинаторных задач.

Введение

В настоящее время высокопроизводительные вычисления стали фактически синонимом параллельных вычислений. Архитектура кластерных мульти-ЭВМ доминирует — в рейтинге 500 наиболее производительных ЭВМ (<http://www.top500.org/>), на данный момент кластеры составляют более 60 %. Важнейшим достоинством кластеров является их относительная дешевизна (на единицу пиковой производительности).

В большинстве программ для кластерных компьютеров используется для обмена данными между процессами метод рассылки сообщений. Основные идеи этого метода были предложены еще в конце 1960-х гг., а в начале 1990-х гг. использовались несколько различных и несовместимых систем рассылки сообщений. Необходимость в обеспечении мобильности и стандартизации привели к созданию MPI (*Message Passing Interface*) [1]. Программный интерфейс MPI представляет собой

описание точного формата вызовов подпрограмм и смысла этих подпрограмм, составляющих библиотеку функций для использования при программировании на языках C или ФОРТРАН. Реализации MPI основываются на открытом стандарте, разработанном открытым образом (общедоступен не только сам стандарт, комментарии-рекомендации к нему, но и рабочие документы MPI-форума). MPI предоставляет средство взаимодействия задач, т. е. вычислений, выполняющихся в разных адресных пространствах.

Стандарт исключает какое-либо скрытое (не включенное в предоставленный программисту интерфейс) взаимодействие ветви параллельного процесса со средой MPI. Это позволяет эффективно переносить программы с одной реализации MPI на другую. Базовой реализацией MPI является MPICH. При этом широко распространены коммерческие продукты, ориентированные на аппаратные особенности архитектуры кластера.

В этой работе рассматриваются вопросы, связанные с разработкой программ для решения логико-комбинаторных задач. Несмотря на ограниченность класса задач, анализ возникающих в этом случае проблем дает представление о проблеме распараллеливания для всего класса задач, решаемых методом перебора. Очень сложно создавать и отлаживать даже последовательные программы, а параллелизм вносит свой еще более высокий уровень сложности. В самых простых с виду параллельных программах обнаруживаются иной раз фатальные ошибки. Ограниченность круга разработчиков можно объяснить отсутствием развитой методологии предотвращения или обнаружения ошибок в параллельных программах. Даже отладка параллельных программ требует методов и инструментов, значительно отличающихся от последовательного случая. Литера-

тура по методам разработки параллельных программ огромна. Эта работа не претендует на роль обзора по технологии параллельного программирования, ее цель — рассказать об опыте разработки важного для практики класса программ.

Комбинаторно-логические задачи характеризуются тем свойством, что для их решения существует универсальный алгоритм, состоящий из полного перебора пространства решений. Так как это пространство всегда конечно, такой перебор в принципе возможен. Решение практических комбинаторно-логических задач требует значительных вычислительных ресурсов. Например, задача нахождения кратчайшего пути в графе, содержащем 100 вершин, методом полного обхода пространства решений требует 25×10^{136} лет работы ЭВМ производительностью 12,5 терафлопс (типа кластерного компьютера СКИФ-1000).

Специфика отладки параллельных программ

Ввиду сложности параллельного программирования MPI программы нуждаются в отладке. Отладку MPI программы можно выполнять с использованием следующих средств.

- Для отладки MPI-программ имеются отладчики классического типа. Такой отладчик выполняет все процессы MPI-программы в режиме отладки. Имеется несколько таких параллельных отладчиков, среди них очень известный коммерческий отладчик *Totalview*. Свободно доступный отладчик *gdb* не имеет в настоящее время специальных средств отладки MPI-программ, однако он может использоваться как процессор отладки в комбинации с клиентской отладочной программой, которая поддерживает MPI, например *mpigdb*.
- Второй подход состоит в том, чтобы использовать отладочную версию библиотеки MPI (например, *mpich*). Эта версия отладки используется не для того, чтобы перехватить внутренние ошибки в библиотеке MPI, а для обнаружения неправильного использования MPI, такого, как ошибки соответствия типов данных в посылке и при получении сообщений.
- Инструменты типа MPI-CHECK ориентированы на программы определенного языка (для MPI-CHECK — это ФОРТРАН) и выполняют проверку типов аргументов или обнаруживают такие некорректности как тупики.

Недостатки этих инструментов в том, что они дороги, требуют модификации исходного текста или используют синтаксические анализаторы языка программирования, ограничены в отношении используемой платформы и языка программирования. Кроме того, они не предотвращают неправильное использование MPI и плохо помогают анализировать ситуацию после того, как неправильное программирование привело к ошибке типа нарушения сегментации.

Опыт отладки параллельных программ (в частности, для решения логико-комбинаторных задач) показывает, что традиционные отладчики и методы отладки последовательных программ не подходят для отладки параллельных. Причина в том, что параллельные программы в целом отличаются от последовательных по методу разработки и модели программы. Кроме того, программы для решения логико-комбинаторных задач имеют особенности в методе распараллеливания.

Наиболее частыми синдромами (внешними признаками) неправильной работы параллельной программы являются тяжелые отказы (типа нарушения сегментации) или "зависание" программы. В обоих случаях классические отладчики не позволяют установить причину ошибки, так как эти синдромы вызваны нарушениями в организации взаимодействия (зависание — как правило, синдром *тупика*, нарушения сегментации, — следствие неправильной работы с буферами каналов), а не ошибками в организации последовательности вычислений, вызванными неаккуратной реализацией алгоритма решения задачи. Тупик — это явление взаимной блокировки процессов.

Хотя параллельное вычисление в основном является дополнительным понятием для последовательного вычисления, они оба имеют общую основу. Любое вычисление предполагает решение задачи путем применения плана (*программы*) действий, выбранных из предопределенного набора, причем каждое из действий состоит в изменении среды вычислений детерминированным способом. При распараллеливании алгоритма для кластерного компьютера требуется декомпозировать проблему на ряд почти независимых последовательных задач, которые могут выполняться одновременно. В последовательном алгоритме задано **как** должно интерпретироваться действие в контексте данной программы. В параллельном алгоритме нужно указать не только **как**, но также и **когда** и **где** действие может (или не может) быть применено.

Гонки между процессами. Явление гонок или состязаний процессов состоит в том, что результат вычислений в информационно связанных между собой процессах зависит от их относительных скоростей выполнения. Программа для кластерного компьютера представляет собой ряд последовательных процессов, которые иногда взаимодействуют, но большей частью выполняются независимо. Процессы программы написаны с учетом условия, что все процессы выполняют асинхронно, т. е., если один процесс нуждается в информации от другого, он приостанавливается для ожидания такой информации, и это взаимодействие не затрагивает другие процессы. Вследствие асинхронности между параллельно выполняющимися процессами возникают явления *гонок* или *тупиков*. Возможность гонок между процессами в параллельных программах с использованием MPI представляет одну из главных проблем их разработ-

ки. Хотя программы с гонками процессов могут быть правильными, имеются алгоритмы, в которых их нужно избегать. Ситуация возникновения гонок не может быть легко воспроизведена, потому что при каждом пуске программа может вести себя по-разному. В любом случае программист должен знать места в программе, где имеется возможность гонок, чтобы решить, допустимы ли они или имеют потенциальную опасность. Очевидно, информацию о наличии гонок трудно получить с применением традиционных отладчиков. Выполнение по шагам кардинально меняет всю динамику протекающих процессов в отлаживаемой программе.

Отладка параллельных программ. Единственным прямым методом снятия данных о динамике выполнения ветвей параллельной программы является вставка программных фрагментов, фиксирующих физическое время начала выполнения этого фрагмента (метод временной трассировки). В этом случае используется старый способ отладки методом включения отладочных сообщений. MPI имеет специальные средства для этой цели. Однако ценность информации, получаемой таким способом, существенно зависит от мест вставки средств временной трассировки. Можно получать информацию о моментах начала и окончания всех операторов MPI с помощью таких средств, как MARMOT [2]. Однако индивидуально выбранные места для вставки позволяют лучше представлять динамику взаимодействия процессов. Временная трассировка всех операторов MPI с помощью таких средств, как MARMOT, дает слишком много ненужной информации. Метод ручной вставки трассирующих сообщений не требует применения специальных инструментов и при умелом применении вполне заменяет автоматизированные инструменты временной трассировки.

При отладке часто встречаются случаи, когда ошибки проявляются только однажды в сотнях прогонов или только на некоторых сочетаниях скоростей процессов. В этом случае альтернативой отладке являются неформальное исследование параллельного алгоритма. Оно основано на применении интуиции и опыта в надежде понять причины возникновения ошибок. К сожалению, такое исследование труднее, чем это кажется на первый взгляд, и почти невозможно для относительно сложных алгоритмов.

Предотвращение ошибок в параллельных вычислениях

Природа ошибок в параллельной программе. Процесс разработки параллельной программы очень трудоемок, хотя на момент ее создания, как правило, уже имеется отлаженная реализация ее последовательного прототипа. При разработке этого прототипа интенсивно и с большим эффектом были использованы отладчики классического типа. Однако параллельная программа обычно все рав-

но требует отладки. Трудности усугубляются тем, что часто параллельная программа разрабатывается на машине с одной архитектурой, а ее практическое применение проводится на другой ЭВМ, отличной от первой, но при этом более мощной. Параллельный вариант отличается от последовательного как минимум наличием механизма взаимодействия одновременно работающих ветвей программы. Причины ошибок в организации последовательного вычисления связаны с неаккуратной реализацией алгоритма решения задачи. Отсутствие системы при распараллеливании алгоритма есть главная причина возникновения ошибок взаимодействия параллельных процессов.

Методика предотвращения ошибок в программах. Такая методика является неотъемлемой частью существующей практики программирования. И хотя существует достаточное число работ о правильных методах программирования и о поиске ошибок в программах, надо сказать, что правильность даже более простой последовательной программы полностью зависит от квалификации программиста. Методики предотвращения ошибок в традиционном программировании призваны устранять влияние невнимательности программиста. Вследствие другой природы ошибок, их предотвращение в параллельных программах, очевидно, должно быть основано на других принципах. Ясно, что если выполнять распараллеливание полностью автоматически, то полученная в результате программа будет безошибочной в части взаимодействия параллельных процессов. Разработка формализованной методики распараллеливания представляет собой решение проблемы трансформации одной модели программы в другую.

В программах обработки данных основными источниками массового параллелизма являются программные циклы. Распараллеливанию таких программ посвящено большое число работ, результаты которых нашли применение в распараллеливающих компиляторах. Для того, чтобы этот метод работал, длины циклов должны быть константами. Для этих программ характерна возможность статического распределения и таких ресурсов как процессоры. Программы для решения логико-комбинаторных задач не могут быть распараллелены таким методом потому, что структура циклов в этих программах изменяется в ходе вычислений. Алгоритм обхода пространства решений обычно рекурсивен, так как первоначальная задача может быть решена методом декомпозиции на задачи того же типа, но меньшей размерности. Рекурсия из соображений эффективности частично заменяется циклическими вычислениями с использованием стека. Структура циклов в таком вычислении полностью зависит от используемых исходных данных.

Однако алгоритмы обхода по дереву легко распараллеливаются другим способом. Полученные в результате декомпозиции задачи могут решаться

параллельно. Проблема состоит в построении *эффективного* параллельного алгоритма, т. е. превосходящего по параметрам задачи лучший последовательный алгоритм. Тривиальный подход к распараллеливанию для почти всех известных комбинаторно-логических задач оказался неэффективным.

Показательным примером может служить задача выполнимости конъюнктивной нормальной формы (КНФ) — *propositional satisfiability* (SAT). Алгоритмы ее решения интенсивно исследуются ввиду важности практических применений. Решение этой задачи требуется при проектировании СБИС, в задачах искусственного интеллекта, составлении расписаний и т. д. Регулярно проводятся соревнования программ (*SAT solver competitions*) [6], на которых выявляются при решении искусственных и реальных задач лучшие программы. Ввиду практической важности решения задачи выполнимости большой размерности, интенсивно исследуются и параллельные алгоритмы ее решения. Несмотря на все усилия на сегодняшний день лучшими программами для решения задачи выполнимости остаются последовательные программы [5]. Причем разрыв в эффективности пока очень велик. Например, неплохой по заложенным принципам работы параллельный решатель при решении одной из задач соревнований решал ее на кластере из 256 процессоров 8 суток, а лучший однопроцессорный вариант решил ее за 18 мин [5].

Объяснение этого факта очевидно: последовательные алгоритмы решения комбинаторно-логических задач исследованы гораздо лучше параллельных, и исследования первых прогрессируют пока быстрее вторых. Следует отметить, что для некоторых задач комбинаторно-логического характера разработаны эффективные параллельные алгоритмы [8].

В настоящее время программы для кластерных компьютеров разрабатываются на языках C или ФОРТРАН. Возможно, проблема автоматического распараллеливания решается путем использования более подходящего языка программирования. Иногда утверждается, что для эффективного использования параллельных компьютеров достаточно применения языков программирования, которые не вводят ненужные последовательные взаимозависимости между подвычислениями программы. Это представление особенно принято в среде приверженцев декларативных языков, так как эти языки обеспечивают высокую степень свободы в упорядочении выполнения компонентов программы. Однако имеющийся опыт разработки параллельных программ настоятельно указывает, что такое представление является чрезмерно оптимистическим. Параллелизм требует явного внимания программиста, особенно в случае программирования для кластерного компьютера с использованием интерфейса MPI. Программист

должен явно указать те варианты параллельного выполнения, которые улучшают общее время вычислений. Задача распараллеливания состоит в том, чтобы явно задать эффективный параллелизм в исходном тексте программы. Эта задача проистекает из того факта, что параллельное выполнение является особенностью реализации для повышения эффективности, в то время как декларативное программирование имеет тенденцию подчеркивать денотационное (максимально очищенное от особенностей реализации) представление.

MPI — это программный интерфейс, который позволяет программисту использовать параллелизм и высокую производительность современных суперкомпьютеров. В целях достижения максимальной выразительности и гибкости этот программный инструмент допускает создание структур, которые являются динамическими, сложными и, следовательно, с трудом поддающимися формальному анализу. Плохо структурированное взаимодействие — причина многих трудностей параллельного программирования. Программист, создавая программу с использованием MPI, должен принимать решения низкого уровня, фиксируя их в тексте программы и выполняя при этом всю необходимую оптимизацию. Вместе с тем, существующие формальные модели взаимодействия [7] часто ограничивают выразительность алгоритма, используя только некоторые из доступных структур взаимодействия. Сужение выразительности компенсируется возможностью формального анализа алгоритма, что ведет к лучшим показателям эффективности программирования за счет использования формальных методов при разработке, проверке, преобразовании и отладке параллельных алгоритмов.

Взаимодействие параллельных процессов

Любые задачи, которые выполняются в кооперации множеством объектов, независимо от того являются ли эти объекты людьми или компьютерами, требуют некоторой формы синхронизации. Как показывает опыт каждодневной жизни, синхронная работа в сотрудничестве облегчается, когда коммуникации между участниками происходят постоянно и без задержек, например, в случае, когда участники сотрудничества — это люди и они находятся в одном помещении. Когда коммуникация ограничивается специальным синхронизатором, сотрудничество становится более трудным.

Синхронизация также связана с упорядоченностью, которая зависит от того, как хорошо участники решения задачи (процессоры кластерного компьютера в рассматриваемом случае) дополняют друг друга в процессе совместной работы. И синхронизация, и упорядоченность влияют на *масштабируемость* — параметр, который характе-

ризует увеличение производительности в зависимости от числа занятых процессоров. При идеальной масштабируемости увеличение числа процессоров вдвое должно приводить к выполнению задачи, по крайней мере, в 2 раза быстрее. Но так бывает редко и часто имеется верхний предел числа процессоров, выше которого использование дополнительных процессоров не уменьшает время, требуемое для выполнения задачи. Синхронизацию принято считать понятием технологии программирования, обеспечивающим в асинхронно выполняющихся информационно связанных процессах информационное взаимодействие. В этом подходе считается, что каждый процесс имеет свои внутренние логические часы, и проблема синхронизации состоит в обеспечении одинаковости хода этих часов [9]. На данную тему опубликовано большое число работ, хотя применимость подхода зависит от возможности до исполнения программы определить отношение частичного порядка между ее подвычислениями.

МРІ программы — это реактивные системы, в которых система реализации МРІ является окружающей средой. Реактивные системы имеют следующие особенности: *время отклика* задается их окружением; их поведение *детерминировано*.

С развитием параллельных аппаратных средств был разработан ряд языков, которые обеспечивают спецификацию и управление параллелизмом [7]. Эти языки нашли ограниченное применение вследствие множества недостатков, главные из которых — зависимость от архитектуры машины, низкая масштабируемость и отсутствие средств автоматизации программирования и отладки.

Следовательно, и сегодня для программистов остается актуальным создание формализма для разработки, реализации и отладки параллельных программ. К настоящему времени известны два подхода к решению этой проблемы.

1. Синхронная параллельная модель. *Параллелизм допустим только в пределах данного "шага" решения.*

2. Асинхронная параллельная модель. *Программа представляет собой ряд последовательных процессов, которые иногда взаимодействуют, но большей частью выполняются независимо.*

Синхронные модели требуют расчленения задачи на ряд маленьких шагов, которые могут быть выполнены параллельно, в то время как асинхронные модели рассматривают проблему как ряд почти независимых последовательных задач, выполняемых одновременно. Обычно синхронные и асинхронные модели рассматриваются как взаимно исключающие.

На интуитивном уровне понятия "синхронный" и "асинхронный" связаны с представлением о наличии в системе общих часов. Часы — генератор синхросигналов — это низкоуровневое понятие, используемое при проектировании аппара-

туры. На уровне модели, очевидно, этому понятию соответствует понятие *времени*. Синхронная и асинхронная модели алгоритмов основаны на учете длительности действий, в то время как последовательная модель предположений о длительности действий не содержит. И в синхронной и в асинхронной модели о длительности вычислительных действий предположений не делается. Различие этих моделей связано с предположениями о длительности действий, обеспечивающих коммуникацию.

В противоположность синхронным, асинхронные модели ориентированы на коммуникации между процессами как на средства управления параллельным выполнением. Моделей коммуникации две: обмен сообщениями (как в МРІ) и общая память. Однако очевидно, что независимо от модели, коммуникация в асинхронных языках связана с синхронизацией. Таким образом, ключевой аспект асинхронных языков, в противоположность синхронным языкам, — задание средств синхронизации процессов. Синхронизация в программной модели — реальная проблема, так как без синхронизации отсутствует *детерминизм* выполнения. Хотя набор механизмов синхронизации в большинстве асинхронных параллельных языков весьма богат, они не обеспечивают спецификации детерминизма явно. В асинхронной системе предполагается, что передача сообщения между процессами происходит через канал взаимодействия, который вызывает ненулевую *задержку* (*latency*).

Верификация на модели (Model Checking)

Сегодня используются два подхода к верификации программного обеспечения. Первый подход, *дедуктивный*, представлен такими направлениями исследований, как автоматическое доказательство теорем, использование мультимножеств и графов, а также разнообразных специализированных алгебр. Программная система описывается в рамках некоего формализма, после чего выполняется строгое математическое доказательство обладания данной системой тех или иных свойств. Второй подход — *модельный*, его последователи не стремятся вписать систему в рамки теории, а вместо этого строят модель системы, которую можно рассматривать как автомат. Любое требование к системе проверяется для каждого возможного состояния автомата.

Методы проверки на модели основаны на переборе состояний, в которые может перейти система в зависимости от структуры запросов и ответов ветвей параллельной программы. Перебор управляется условиями, которые сформулированы на языке временной логики и выражают корректные состояния взаимодействий программы. Методы проверки на модели имеют существенное ограничение: рассматриваются лишь системы с

конечным набором состояний (хотя сам моделируемый процесс может быть бесконечным).

Сильные и слабые стороны модельного подхода. Модельных подходов известно по меньшей мере дюжина — конечные автоматы, сети Петри, временные автоматы, логическое описание и т. п. Попробуем перечислить присущие им общие сильные свойства.

Модельный подход поддерживает не только полную, но и частичную верификацию, которая может быть направлена на проверку только одного свойства, абстрагировавшись от менее важных деталей системы. Иными словами, для проведения верификации не обязательно добиваться формализации всех без исключения требований к программе. В отличие от тестирования и использования симуляторов, в модельном подходе не существует такого понятия, как *вероятность обнаружения ошибки*: если ошибка есть, она будет обнаружена за конечное время.

В том случае, когда верифицируемое свойство оказывается нарушенным, в виде контрпримера предоставляется диагностирующая информация. Процесс проверки моделей не требует ни ручного управления со стороны пользователя, ни высокого уровня профессионализма. Имея модель, можно автоматически проверять на ней необходимые свойства. Процесс проверки интегрируется в стандартный цикл проектирования, позволяя, как показывает практика, уменьшить время создания приложений с учетом времени разработки самой модели.

Однако у модельного подхода есть и слабые стороны. Верификация осуществляется по модели, а не по реальной системе, поэтому ценность полученного результата напрямую зависит от корректности модели, что требует высокого уровня подготовки персонала, создающего модели программ.

Преобладает ориентация на приложения, в которых главную роль играет *поток управления*, а не *поток данных*, так как данные имеют тенденцию принимать значения из бесконечных множеств. Такая ориентация уменьшает возможности универсального применения, однако для алгоритмов решения комбинаторно-логических задач характерна именно архитектура "потока управления". Нет гарантий полноты: проверяются только те свойства, которые указаны явно.

Построение моделей и формулировка требований требуют высокого уровня знаний и умения их применять. Результаты могут вводить в заблуждение (верификатор — тоже программа и тоже может ошибаться, модель может содержать ошибку и т. п.); правда, основные процедуры проверки моделей формально доказаны с помощью пакетов автоматического доказательства теорем).

В качестве примера показательно исследование алгоритма разделения множеств, разработанного Э. Дейкстрой. Алгоритм может быть реали-

зован с применением интерфейса MPI, текст программы составляют около десятка строчек. Программа много обсуждалась в публикациях, ее частичная корректность была доказана разными авторами, однако формальными методами были обнаружены конкретные ситуации, когда исполнение алгоритма приводит к ошибке [10]. Этот пример убедительно показывает, что простое тестирование, без формального обоснования правильности программы, не в состоянии обеспечить правильность параллельной программы.

Разработка автоматического экстрактора модели из программы MPI. В настоящее время выполнено значительное число исследований в области автоматического извлечения модели из программ, написанных на языках типа C или JAVA, и большая часть этих результатов можно перенести на программы MPI. Однако практического инструмента, применимого к программам MPI, до сих пор не существует [4].

Заключение

Методы верификации на модели имеют значительный потенциал для поиска ошибок в MPI-программах, позволяя проверять некоторые свойства, присущие "правильным" параллельным программам. Такие программы трудно написать без ошибок и чрезвычайно трудно отлаживать. Ряд особенностей MPI, например, непредсказуемый порядок буферизации сообщений, делают множество состояний таких программ особенно большими и таким образом представляют существенные препятствия успешному применению методов верификации на модели.

Список литературы

1. **Message Passing Interface Forum.** MPI: A Message-Passing Interface standard, version 1.1. <http://www.mpi-fomm.org/docs/>. 1995.
2. **Krammer B., Muller M. S., Resch M. M.** MPI Application Development Using the Analysis Tool MARMOT // Proc. of International Conference on Computational Science. 2004. P. 464—471.
3. **Harel D., Pnueli A.** On the development of reactive systems // In Logic and Models of Concurrent Systems. NATO Advanced Study Institute on Logic and Models for Verification and Specification of Concurrent Systems. Springer Verlag, 1985. P. 477—498.
4. **Siegel S. F., Avrunin G. S.** Modeling wildcard-free MPI programs for verification // Proc. of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (Chicago, IL, USA, June 15—17, 2005) / PPOPP'05. New York: ACM Press, 2005. P. 95—106.
5. **Singer D.** Parallel Resolution of the Satisfiability Problem: A Survey // Parallel Combinatorial Optimization / Ed. by El-Ghazali Talbi, Wiley-Interscience, 2006.
6. **The SAT solver competitions.** <http://www.satcompetition.org>.
7. **Aggarwal A., Chandra A., Snir M.** Communication complexity of PRAMs // Theoretical Computer Science. 1990. 71 (1): 3—28.
8. **Торопов Н. П.** Параллельные логико-комбинаторные вычисления в среде MPI // Информатика / ОИПИ НАН Беларуси. 2005. № 3. С. 82—90.
9. **Lamport L.** Time, Clocks, and the Ordering of Events in a Distributed System // Comm. ACM. July 1978. Vol. 21. N 7. P. 558—564.
10. **Карпов Ю. Г.** Анализ корректности параллельной программы разделения множеств // Программирование. 1996. № 5.

Е. Г. Князев, Д. Г. Шопырин, канд. техн. наук,
Санкт-Петербургский государственный
университет информационных технологий,
механики и оптики

Автоматизированная классификация изменений программного кода методами многомерного статистического анализа

Одной из основных мер общего контроля состояния программной системы является постоянный обзор изменений исходного кода. Однако эта деятельность связана с анализом большого потока информации. Представлен метод автоматизированной классификации изменений программного кода, основанный на многомерном статистическом анализе. Автоматизированная классификация изменения программного кода позволяет увеличить производительность эксперта при выполнении задач, связанных с контролем состояния и анализом истории программных систем.

1. Постановка задачи

При промышленной разработке программного обеспечения часто вычисляются группы разработчиков с явным техническим лидером, например, старшим или ведущим программистом. В обязанности технического лидера обычно входит не только общий контроль состояния программной системы, но и обучение менее опытных разработчиков на примере ошибок, которые они совершают в процессе разработки. Одной из основных мер общего контроля состояния программной системы является постоянный обзор изменений исходного кода. Однако технический лидер зачастую сталкивается с очень большим потоком изменений, который зависит от размеров команды разработчиков и используемого цикла разработки. Например, в случае команды из пяти разработчиков и использования современных *гибких (agile)* процессов разработки [1] можно ожидать, что техническому лидеру придется просматривать как минимум 25 изменений ежедневно.

Автоматизированная классификация изменений, в качестве вспомогательного инструмента, позволяет увеличить производительность эксперта при выполнении задач, связанных с анализом истории программных систем. В частности, использование автоматизированной классификации позволяет отфильтровать несущественные для эксперта изменения системы. Например, при анализе

текущего состояния проекта эксперт может выделить изменения, которые привели к реализации новой функциональности и сосредоточиться на них.

Также автоматизированная классификация изменений может применяться для следующих задач:

- *обеспечение возможности запрета изменений на определенных этапах разработки.* К примеру, на этапе стабилизации программного кода продукта может быть автоматизирована проверка запрета реализации новой функциональности;
- *контроль процесса разработки программного обеспечения с помощью анализа соотношений вносимых изменений.* Автоматизированная классификация изменений помогает оценивать скорость и качество разработки программного продукта по количественному соотношению их типов. Так, преобладание исправлений ошибок на стадии реализации новой функциональности будет свидетельствовать о необходимости введения стадии стабилизации вместо продолжения фазы развития программного продукта.

В настоящей статье предлагается метод классификации изменений программного кода, который позволяет автоматически проводить разделение семантически различных изменений на основе значений метрик исходного кода.

В разделе 2 данной статьи приводится краткий обзор методов *MSR (Mining Software Repositories, анализ исторических хранилищ программного кода)* [2], позволяющих решать задачи классификации изменений путем анализа изменений исходного кода. В разделе 3 описывается предложенный метод классификации изменений. В разделе 4 рассматривается пример практического применения инструмента классификации изменений при разработке программного продукта *Navi-Manager* в компании "Транзас Технологии".

2. Предшествующие исследования

Среди существующих методов классификации изменений можно выделить следующие группы [3]:

- *неформальные методы* — такие, как автоматизированная классификация изменений посредством анализа комментариев [4, 5];
- *методы анализа синтаксиса изменений* — такие, как эвристическое сравнение синтаксических деревьев версии [6] и анализ разницы версии с помощью встраиваемых в исходный код тегов [7].
- *методы, основанные на data mining* — такие, как метод кластеризации исходного кода на основе метрик [8].

Неформальные методы. В работе [4] предложен метод автоматизированной классификации, основанный на анализе комментариев к изменениям кода. В комментариях осуществляется поиск слов, специфичных для каждого из типов изменений. Если вхождение слова найдено, изменение классифицируется как принадлежащее группе, описываемой этим словом. Например, слова "*ис-*

правлено", "ошибка" характеризуют группу исправлений ошибок, а слова "добавлено", "реализовано" — группу добавления новой функциональности. Далее сравниваются результаты автоматизированной и экспертной оценок для тестового набора данных. При таком подходе согласованность результатов достигает 70 %.

Недостатком данного метода является то, что он основывается на анализе комментариев. Однако разработчики не всегда исчерпывающе и корректно описывают проведенные ими модификации кода. Вообще, содержание комментария субъективно, и текст, написанный человеком, может быть интерпретирован различным образом. Поэтому на практике не всегда корректно судить о содержании изменений, основываясь лишь на комментариях к ним.

Методы анализа синтаксиса изменений. В работе [6] с помощью эвристического алгоритма сравнения синтаксических деревьев, построенных для последовательных версий исходных файлов, строится синтаксическая разница между версиями. Затем вычисляется, сколько функций и вызовов было добавлено, модифицировано, сколько условных выражений было изменено и т. д.

В работах [7, 9] анализируется разница версий с помощью тегов, предварительно встроенных в код. Благодаря тегам обнаруживаются добавленные, измененные и удаленные синтаксические вхождения и вычисляются типы синтаксических изменений. В результате работы метода даются ответы на вопросы: добавились ли новые методы в определенный класс, есть ли изменения в директивах препроцессора, было ли модифицировано условие любого условного оператора и т. д.

Недостатками методов анализа синтаксиса изменений является то, что они в большинстве своем сложны и зависят от конкретного языка программирования. Применение таких методов оправдано только для анализа простых изменений. В случае, если структура изменения сложна, что часто встречается на практике, метод не позволит дать однозначного ответа на вопрос о его типе.

Методы, основанные на data mining. На данный момент авторам неизвестны методы автоматизированной классификации изменений программного кода, основанные *data mining* (в дословном переводе — "добыча данных"). Однако многомерный статистический анализ используется для других смежных задач. Например, в работе [8] предложен метод анализа архитектуры программной системы на основе кластеризации ее компонентов связности. Для множества подпрограмм исходного кода вычисляются метрики связности. Затем значения этих метрик кластеризуются. Каждый кластер представляет собой отдельный модуль, составленный из сильно связанных компонентов, в то время как сам модуль слабо связан с остальными модулями. В результате, с помощью оценивания

числа кластеров в системе можно определить степень ее модульности. Модульность, в свою очередь, — это показатель качества системы.

Принципиальным отличием предложенного в настоящей работе метода является применение анализа *изменений* исходного кода, а не только текущего состояния системы, как в работе [8]. Это позволяет, распространив анализ кода во временную плоскость, исследовать *процесс развития* программ. Предложенный метод дает возможность решать задачу автоматизированной классификации изменений программного кода без применения синтаксического анализа. Также предложенный метод позволяет выявлять скрытые зависимости, которые не могут быть обнаружены простым анализом исходного кода.

3. Метод кластеризации метрик изменений

Метод автоматизированной классификации изменений, предложенный в настоящей работе, относится к *методам data mining*. Он основан на кластеризации значений метрик изменений исходного кода с помощью метода *Мак-Куна* [10, 11]. В результате проводится разбиение множества изменений на заданное число кластеров, каждый из которых соответствует определенному типу изменений.

Формализация задачи. Изменение кода *программной системы* δ трактуется в работе как отображение множества исходных данных S в другое множество, модифицированных данных S^* :

$$\delta : S \xrightarrow{\delta} S^*.$$

В некоторых современных системах хранения исходного кода каждому состоянию исходного кода последовательно сопоставляется неотрицательное целое число r , которое называется *ревизией* или *версией программного кода*. Поэтому каждое изменение исходных данных можно описать следующим образом:

$$\delta_r : S_r \xrightarrow{\delta_r} S_{r+1}.$$

Каждое изменение δ_r может быть отнесено экспертом к некоторому множеству типов изменений t_r , где $t_i \in T$ — тип изменения δ_r . При этом T представляет собой множество типов изменений, специфичное для каждого конкретного проекта. Состав множества T определяется экспертом в зависимости от специфики проекта. Во множество T обычно входят такие типы изменений, как *реализация новой функциональности*, *рефакторинг*, *исправление ошибки* и т. д. [12].

Задача отнесения изменения к тому или иному типу изменений трудоемка и требует высокой квалификации эксперта, так как нет четких критериев оценки типа изменения. Введем функцию интерпретации изменений I , отображающую множество изменений $\{\delta_r\}$ во множество их типов $\{t_i\}$:

$$I : \{\delta_r\} \xrightarrow{I} \{t_1, t_2, \dots, t_n\}.$$

Предлагается автоматизировать процесс выделения типов изменений с помощью кластеризации метрик изменений. В процессе кластеризации строится множество кластеров изменений C такое, что каждое изменение δ_r относится к некоторому кластеру $c_j \in C$.

Введем функцию автоматизированной классификации изменений I_A , отображающую множество изменений $\{\delta_r\}$ во множество их типов $\{t_i\}$:

$$I_A: \{\delta_r\} \xrightarrow{I_A} \{t_1, t_2, \dots, t_n\}.$$

Здесь функция автоматизированной классификации I_A есть композиция функций кластеризации I_C и интерпретации кластеров I_T :

$$I_A = I_C \circ I_T;$$

$$I_C: \{\delta_r\} \xrightarrow{I_C} \{c_1, c_2, \dots, c_m\};$$

$$I_T: \{c_1, c_2, \dots, c_m\} \xrightarrow{I_T} \{t_1, t_2, \dots, t_n\}.$$

Функция кластеризации I_C отображает множество изменений во множество кластеров. Функция интерпретации кластеров I_T отображает множество кластеров C в множество типов изменений T .

Функция кластеризации I_C может быть построена с помощью метода многомерной кластеризации Мак-Куина [10, 11]. Кластеризацию изменений будем осуществлять на основе некоторых метрик изменений $M'_r \delta_r$. Определим понятие метрики изменения через понятие метрики программного обеспечения.

Метрика программного обеспечения (software metric) — это мера M , позволяющая получить числовое значение некоторого свойства программного обеспечения S или его спецификаций [13, 14], например, число строк исходного файла, цикломатическая сложность [15], число ошибок на строку кода, число классов и интерфейсов, связность и др.

Тогда метрику изменения программного обеспечения можно определить как разность значений метрики измененного кода MS_{r+1} и метрики исходного кода MS_r :

$$M'_r \delta_r = MS_{r+1} - MS_r. \quad (1)$$

Выберем набор метрик программного обеспечения $\bar{M} = \langle M_1, M_2, \dots, M_p \rangle$. Тогда для каждого изменения δ_r можно построить набор метрик изменения:

$$\bar{M}'\delta_r = \langle M'_1 \delta_r, M'_2 \delta_r, \dots, M'_p \delta_r \rangle.$$

Тогда $\bar{M}'\delta_r$ — это точка в k -мерном пространстве кластеризации. Мерой расстояния между точками в этом пространстве выберем евклидово расстояние ρ :

$$\begin{aligned} \rho(\langle x_1, x_2, \dots, x_k \rangle, \langle y_1, y_2, \dots, y_k \rangle) = \\ = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_k - y_k)^2}. \end{aligned}$$

Теперь разбиение на кластеры может быть задано следующим образом:

$$\begin{aligned} C = \{c_1, c_2, \dots, c_m\}, c_j = \\ = \{\delta_r, \delta_g | \rho(\bar{M}'\delta_r, \bar{M}'\delta_g) < \rho_{\min}\}, \end{aligned} \quad (2)$$

где ρ_{\min} — величина, определяющая меру близости для включения объектов в один кластер.

Алгоритм кластеризации изменений. Пусть известно число кластеров m , выбран набор метрик \bar{M} и мера расстояния ρ между точками пространства кластеризации принята евклидовой. В соответствии с методом многомерной кластеризации Мак-Куина [10, 11], алгоритм кластеризации изменений следующий:

1. Выполнить начальное разбиение множества объектов $\{\delta_r\}$ случайным образом:

$$C^0 = \{c_1, c_2, \dots, c_n\}, c_i = \{\delta_r | \delta_r \notin c_j, j \neq i\}.$$

2. Принять номер итерации $l = 1$.

3. Определить центры кластеров cc_i по формуле

$$cc_i = \frac{\sum_r [\delta_r \in c_i] \bar{M}'\delta_r}{\sum_r [\delta_r \in c_i]}.$$

4. Обновить множества распределения объектов по кластерам $C^l = \{c_i\}$:

$$c_i = \{\delta_r | \rho(\bar{M}'\delta_r, cc_i) = \min \rho(\bar{M}'\delta_r, cc_j)\}.$$

5. Проверить условие $\sum_i \|c_i^l \Delta c_i^{l-1}\| = 0$, где

Δ — операция взятия симметрической разности множеств: $A \Delta B = (A \cup B) \setminus (A \cap B)$. Если условия выполнены, то завершить процесс, иначе перейти к шагу 3 с номером итерации $l = l + 1$.

Приведенный алгоритм позволяет автоматически разбить множество изменений на кластеры. В каждый кластер группируются наиболее схожие друг с другом изменения.

Интерпретация результатов кластеризации.

Подбор подходящего числа кластеров m и построение функции интерпретации кластеров I_T проводится экспертом на основе выборочного анализа изменений, принадлежащих каждому кластеру. Эти задачи значительно менее трудоемки, чем исходная задача, так как на практике имеет смысл различать лишь небольшое число типов изменений.

В процессе построения функции интерпретации кластеров I_T экспертом анализируются изменения исходного кода и комментариев, сопровождающий изменение. В результате устанавливается, какому из типов t_j соответствует данный кластер c_j . При невозможности сопоставления кластера изменений экспертному типу следует повторно обратиться к выбору метрик для кластеризации.

Таблица 1

Метрики, используемые в процессе кластеризации

Метрика	Описание
eLOC	Эффективное число строк кода
CC	Цикломатическая сложность (число независимых путей в графе исполнения)
IC	Интерфейсная сложность (общее число параметров во всех методах)
C/S	Число классов и структур

4. Пример практического применения метода автоматизированной классификации

Представленный в работе метод успешно опробован на практике. Реализован программный инструмент кластеризации ревизий репозитория системы контроля версий *Subversion*. Метрики исходного кода, написанного на любом из языков C, C++, C#, *Java*, вычисляются с помощью инструмента *MSquared Resource Standard Metrics* [17]. Разработанный инструмент внедрен в компании "Трансас Технологии" и доказал эффективность практического применения предложенного метода.

В качестве иллюстрации метода кластеризации метрик изменений в текущем разделе приводятся результаты применения метода на проекте *Navi-Manager*. *Navi-Manager* — это система слежения за флотом, состоящая из серверной и клиентской частей. Система преимущественно реализована на языке C#. Общее число строк кода — более 100 000.

В качестве исходных данных анализа случайным образом выбрано 13 изменений, выполненных в проекте *Navi-Manager* за шесть месяцев разработки, и проведена экспертная классификация этих изменений по типам. Экспертом выделены следующие типы изменений в наборе: добавление новой функциональности, удаление функциональности, исправление ошибки, рефакторинг, косметические изменения.

Первоначальный выбор набора метрик для автоматизированной классификации осуществляется на основе эмпирических представлений эксперта о способности разделения экспертных типов изменений алгоритмом кластеризации по различиям значений их метрик. Выбранные метрики, предположительно связанные с различием экспертных типов изменений, приведены в табл. 1 [0].

В процессе автоматизированной кластеризации было построено множество из четырех кластеров, которые с помощью эксперта были сопоставлены выделенным типам изменений. Результаты этого сопоставления приведены в табл. 2.

В табл. 2 приведены:

- ревизия изменения;
- комментарий к изменению;
- тип изменения, который был присвоен изменению в процессе оценки экспертом;
- кластер, который был сопоставлен изменению в процессе кластеризации;
- интерпретация кластеров экспертом.

В табл. 2, 3 использованы следующие сокращения:

- "Удал." — удаление кода;
- "Реф." — рефакторинг;
- "Испр." — исправление ошибки;
- "Нов." — новая функциональность.

Для оценки согласованности автоматизированной и экспертной классификации в работе используется коэффициент Кохена [20, 21]. Коэффициент Кохена представляет собой меру согласия, с которой два эксперта конкурируют в своих сортировках N элементов по k взаимно исключающим категориям. Эксперта в данном контексте может представлять человек или множество людей, которые коллективно распределяют N элементов, или некоторый алгоритм, который распределяет элементы на основе некоторого критерия.

Коэффициент Кохена вычисляется следующим образом. В табл. 3 размером $(n + 1) \times (n + 1)$ выписываются результаты экспертной и автома-

Таблица 2

Результаты автоматизированной и экспертной классификации изменений из проекта *Navi-Manager*

Ревизия	Комментарий	Тип изменений	Номер кластера	Интерпретация
12883 18281	Перенос файлов между проектами, удаление ненужных Уменьшено дублирование кода	Удал. Удал.	1	Удал.
17135 18048 16743	Обобщена логика FindOrCreateUserSource Рефакторинг работы с интервалами Обобщена обработка GlobeWireless	Реф. Реф. Реф.	2	Реф.
17613 17929 17970 18273	Увеличен таймаут PositionDiscarder до 5 мин Исправлено несколько логических ошибок в работе TrackAgent Переименован ITrackFetch.Interval в DestInterval Поправлена ошибка при проверке флагов рисования элемента меню	Испр. Испр. Реф. Испр.	3	Испр.
15115 15191 16580 16899	Поправлена установка vessel.LastReportTime Добавлены блоки try..catch в методы OnStart, OnStop сервиса Добавлен перенос max LastReportTime, LastSourceId в логику объединения данных объекта Добавлена инверсия цвета текста в колонку L в Vessel List View	Испр. Нов. Нов. Нов.	4	Нов.

Таблица 3

Соответствие автоматизированной и экспертной классификации измерений *Navi-Manager*

Экспертная классификация	Автоматизированная классификация				
	Удал.	Реф.	Испр.	Нов.	Сумма
Удал.	2				2
Реф.		3	1		4
Испр.			3	1	4
Нов.				3	3
Сумма	2	3	4	4	13

тизированной классификации (число кластеров должно быть равно числу экспертных типов).

Элемент табл. 3 p_{ij} — число изменений, принадлежащих к типу t_i автоматизированной классификации и одновременно к типу t_j экспертной классификации. На главной диагонали расположены величины, равные числу изменений, для которых совпадает результат автоматизированной и экспертной классификации. В строке "Сумма" приведено распределение частот изменений по типам автоматизированной классификации. В столбце "Сумма" приведено распределение частот изменений по экспертным типам. Правый нижний элемент табл. 3 содержит общее число анализируемых изменений.

Выражение для расчета коэффициента согласия Кохена следующее:

$$\kappa = \frac{p_{\alpha} - p_e}{1 - p_e},$$

где p_{α} — относительное наблюдаемое согласие между экспертами; p_e — вероятность обусловленности этого согласия случайностью. Если эксперты находятся в абсолютном согласии между собой, тогда $\kappa = 1$. Если же согласие между экспертами отсутствует (но не по причине случайности), тогда $\kappa \leq 1$.

Вычисление коэффициента согласованности автоматизированной и экспертной классификации выполняются по кросс-табуляции наблюдаемого числа наблюдений изменений по каждому типу (табл. 3). Тогда величина p_{α} рассчитывается как отношение суммы элементов табл. 3 по диагонали к общему числу элементов:

$$p_{\alpha} = \frac{\sum p_{ii}}{\Sigma}.$$

Величина p_e вычисляется как отношение суммы диагональных элементов таблицы, полученной случайным распределением по категориям, к общему числу элементов:

$$p_e = \sum_k p_{k+} p_{+k},$$

где $p_{k+} = \frac{\sum_j p_{kj}}{\Sigma}$ и $p_{+k} = \frac{\sum_i p_{ik}}{\Sigma}$ — пропорции элементов по строкам и колонкам от общего их числа.

По результатам эксперимента получено значение каппа, равное 0,79, которое лежит на границе значительной и превосходной степени согласованности двух методов классификации [21].

Заключение

Преимущества предложенного метода состоят в следующем:

- *объективность*: для анализа используется исходный код, а не, например, комментариев, сопровождающий изменение. Оценка по исходному коду адекватна в отличие от классификации комментариев к изменениям, ведь комментарии могут не в полной мере соответствовать характеру изменений [22];
- *настраиваемость*: множество метрик программного кода может выбираться в зависимости от того, по каким аспектам изменений предполагается группировка [13, 14];
- *адаптивность*: при кластеризации задается лишь результирующее число групп. Следовательно, для каждого отдельно взятого проекта предложенный метод позволяет выделить специфичные множества изменений, которые затем интерпретируются как те или иные семантические группы изменений;
- *формальность*: кластеризация осуществляется с помощью формальных методов добычи данных вообще и анализа программных репозиторий, в частности.

Практическое применение метода выявило ряд еще не решенных проблем. Например, возникает проблема выбора числа групп для кластеризации. Эта проблема описана, например, в работе [11]. Задача определения числа кластеров решается опытным путем, а именно, выполнением алгоритма кластеризации над исходными данными с несколькими значениями n . Дополнительно предполагается участие эксперта в процессе интерпретации результатов автоматической кластеризации.

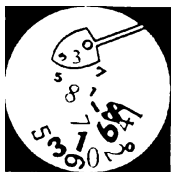
Также стоит проблема *смешанных изменений*, сочетающих в себе разнородные модификации кода. Настоящим методом не всегда возможна корректная классификация таких изменений. Нужно заметить, что наличие смешанных изменений на практике нежелательно и даже вредно. При их наличии усложняется процедура просмотра кода и другая работа с историей программного продукта. Выделение смешанных изменений в отдельный тип в процессе кластеризации — предмет дальнейших исследований.

Решение этих и других проблем является целью дальнейших исследований. Однако уже сей-

час, в результате анализа качества автоматизированной классификации для тестовых выборок, делается вывод о практической способности предложенного метода классифицировать изменения программного кода адекватным образом.

Список литературы

1. **Aoyama M.** Agile software process and its experience // IEEE Computer Society. Proceedings of the 20th international Conference on Software Engineering (Kyoto, Japan, April 19–25, 1998). International Conference on Software Engineering. Washington: DC. 1998. P. 3–12.
2. **Zimmermann T.** Knowledge Collaboration by Mining Software Repositories // Proceedings of the 2nd International Workshop on Supporting Knowledge Collaboration in Software Development (KCSO 2006). Tokyo, Japan, 2006. P. 64, 65.
3. **Kagdi H., Collard M., Maletic J.** Towards a Taxonomy of Approaches for Mining of Source Code Repositories // ACM SIGSOFT Software Engineering Notes, Proceedings of the 2005 international workshop on Mining software repositories MSR '05. St. Louis, Missouri. 2005. P. 1–5.
4. **Hassan A. E., Holt R. C.** Source Control Change Messages: How Are They Used And What Do They Mean? 2004. <http://www.ece.uvic.ca/~ahmed/home/pubs/CVSSurvey.pdf>.
5. **Mockus A., Votta L. G.** Identifying reasons for software change using historic databases // Proceedings of the International Conference on Software Maintenance (ICSM). San Jose, California. 2000. P. 120–130.
6. **Raghavan S., Rohana R., Podgurski A., Augustine V.** Dex: A Semantic-Graph Differencing Tool for Studying Changes in Large Code Bases // Proceedings of 20th IEEE International Conference on Software Maintenance (ICSM'04). Chicago, Illinois, September 11–14, 2004. P. 188–197.
7. **Maletic J. I., Collard M. L.** Supporting Source Code Difference Analysis // Proceedings of IEEE International Conference on Software Maintenance (ICSM'04). Chicago, Illinois. September 11–17. 2004. P. 210–219.
8. **Maitra R.** Clustering Massive Datasets With Application in Software Metrics and Tomography // Technometrics. 1 August 2001. Vol. 43. N 3. P. 336–346.
9. **Collard M. L.** Meta-Differencing: An Infrastructure for Source Code Difference Analysis. Kent State University, Kent, Ohio USA. Ph. D. Dissertation Thesis. 2004.
10. **Баргесян А. А., Куприянов М. С., Степаненко В. В., Холод И. И.** Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP. СПб.: БХВ-Петербург, 2007. 375 с.
11. **Мандель И. Д.** Кластерный анализ. М.: Финансы и статистика, 1988. 176 с.
12. **Фаулер М.** Рефакторинг: улучшение существующего кода. СПб.: Символ-Плюс, 2003. 432 с.
13. **Орлов С. А.** Технологии разработки программного обеспечения: Учеб. для вузов. СПб.: Питер. 2004. 528 с.
14. **Липаев В. В.** Выбор и оценивание характеристик качества программных средств: Методы и стандарты. М.: Синтез. 2001. 224 с.
15. **McCabe T. J.** A Complexity Measure // IEEE Trans SE-2, December 1976. N 4.
16. **Collins-Sussman B., Fitzpatrick B. W., Pilato M.** Version Control with Subversion. O'Reilly. 2004. <http://svnbook.red-bean.com/>.
17. **MSquared Resource Standard Metrics.** Code Analysis for Code Reviews and Acceptance. <http://msquaredtechnologies.com/m2rsm/index.htm>.
18. **Navi-Manager Vessel Monitoring System.** <http://www.transas.com/products/shorebased/manager/>. <http://www.transas.ru/products/shorebased/fleet/navi-manager/>
19. **MSquared Resource Standard Metrics Documentation.** http://msquaredtechnologies.com/m2rsm/docs/rsm_metrics_narration.htm.
20. **Cohejn J.** A Coefficient of Agreement for Nominal Scales // Educational and Psychological Measurement. 1960. P. 37–46.
21. **Elemam K.** Benchmarking Kappa for Software Process Assessment Reliability Studies // Technical Report ISERN-98-02, International Software Engineering Research Network. 1998. <http://cite-seer.ist.psu.edu/elemam98benchmarking.html>
22. **Zimmermann T., Wei gerber P., Diehl S., Zeller A.** Mining Version Histories to Guide Software Changes // Proceedings of 26th International Conference on Software Engineering (ICSE'04). Edinburgh, Scotland, United Kingdom, May 23–28. 2004. P. 563–572.



ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

УДК 004.896:621.771

С. М. Кулаков, д-р техн. наук, **В. Б. Трофимов**,
Н. Ф. Бондарь, канд. техн. наук, доц., **С. В. Чабан**

Сибирский государственный индустриальный университет, г. Новокузнецк

Интеллектуальная система распознавания поверхностных дефектов проката

Рассматривается актуальная прикладная задача распознавания дефектов поверхности стального проката на примере рельсов в процессе их производства. Для ее решения предложена комбинированная процедура, объединяющая аппарат искусственных нейронных сетей и динамических экспертных систем.

Введение

Рассмотрим особенности трактовки понятия "распознавание" по работам [1–6]. Согласно [1], задача распознавания образов заключается в классификации изображений ("... образом является изображение") на основе определенных тре-

бований, причем изображения, относящиеся к одному классу образов, обладают относительно высокой степенью близости. Распознавание представляет собой классификацию на множестве признаков, оцениваемых по наблюдаемому изображению. Процесс отбора информативных при-

знаков до сих пор остался процедурой эвристической, зависящей как от предметной области, так и от предпочтений разработчика. Можно также сказать, что классификация образов заключается в отображении пространства признаков в пространства решений. При таком подходе распознавание образов включает две задачи:

- отбор и упорядочивание информативных признаков (геометрических, топологических, вероятностных, спектральных);
- собственно классификация (принятие решения о принадлежности изображения к тому или иному классу на основе анализа значений признаков).

Академик Я. З. Цыпкин [2] утверждает, что основная задача опознавания (узнавания, распознавания) состоит в отнесении предъявляемого объекта к одному из классов. Классы характеризуются тем, что принадлежащие им объекты обладают некой общностью, сходством. То общее, что объединяет объекты в класс, и называют *образом*. Для решения задачи опознавания необходимо первоначально заняться обучением посредством показа образов, принадлежность которых к тому или иному классу известна.

В учебном пособии А. Л. Горелика и В. А. Скрипкина [3] распознавание представляет собой задачу преобразования входной информации (в качестве которой уместно рассматривать некоторые параметры, информативные признаки распознаваемых образов, например, объектов, сигналов, ситуаций, явлений или процессов) в выходную, представляющую собой заключение о том, к какому классу относится распознаваемый образ.

В работе Дж. Ф. Люгера [4] классификация — это определение категории или группы, к которой принадлежат входные значения, а распознавание образов — идентификация структуры или шаблона данных.

К. Фу в книге [5] утверждает, что для распознавания образов более совершенным подходом (чем подход, основанный на сравнении входного образа с эталонами) является классификация, базирующаяся на некотором множестве "отобранных замеров", проводимых на входных образах. Эти "отобранные замеры", называемые *признаками*, предполагаются инвариантными по отношению к обычно встречающимся искажениям и обладающими малой избыточностью. Классификация при этом рассматривается как принятие решения о принадлежности входного образа тому или иному классу.

Р. Дуда и П. Харт в труде, связанном с исследованиями по искусственному интеллекту [6], структуру распознавателя образов рассматривают как модель интеллекта, состоящую из трех частей: датчика, выделителя признаков и классификато-

ра. Датчик воспринимает воздействие объекта и преобразует его в сигналы (первичные данные), удобные для машинной обработки. Выделитель признаков (называемый также рецептором, фильтром свойств, детектором признаков или препроцессором), ориентированный на уменьшение объема обрабатываемых данных, выделяет из первичных данных полезные сведения. Классификатор на основе этих сведений относит объект к одной из нескольких категорий.

Постановка задачи распознавания дефектов рельсов

Дано:

- Множество характерных фрагментов оцифрованных изображений поверхности дефектных и бездефектных участков рельсов.
- Библиотека многовариантно представленных типовых дефектов (волосовина, раскатанный пузырь, раскатанная трещина, плена, рванина, риски).
- Описание действующей системы визуального контроля качества рельсов в ОАО "НКМК".
- ГОСТ 21014—88 "Прокат черных металлов. Термины и определения дефектов поверхности", атласы дефектов [7—9], технологические инструкции производства рельсов и их приемки.
- Описание определяющих факторов, обуславливающих возникновение поверхностных дефектов.
- Методы и алгоритмы выделения информативных признаков и классификации.
- Аппарат ИНС: многослойный перцептрон, алгоритм обратного распространения ошибки, алгоритм Левенберга—Марквардта, алгоритм сопряженных градиентов, алгоритм быстрого распространения.
- Группа специалистов-экспертов, квалифицированно решающих задачу распознавания дефектов неподвижного рельса.
- Методика построения продукционной модели представления знаний: IF (условие), THEN (действие) [10].
- Программный продукт разработки экспертных систем, поддерживающих работу в сети Internet — Exsys CORVID [11].
- Программный продукт Borland Delphi.
- Типовые факторы, искажающие изображение фрагментов рельса (масляные пятна, сдвиг, поворот, вибрация).
- Критерий качества распознавания — отношение числа правильно опознанных объектов к общему их числу в партии.
- Ограничение — диагностические решения должны приниматься с упреждением на время, достаточное для отметки дефектных участков рельса.

Требуется разработать интеллектуальную систему распознавания дефектов рельсов, движущихся по рольгангу, учитывающую заданные ограничения и обеспечивающую максимум правильно опознанных дефектов.

Методика

Предлагается интеллектуальная система распознавания поверхностных дефектов рельсов, объединяющая две технологии — искусственные нейронные сети (ИНС) и экспертные системы (ЭС).

Процедура распознавания поверхностных дефектов рельсов в потоке производства (рис. 1) включает следующие основные операции:

1. Сканирование поверхности рельса с помощью восьми видеокамер и специальной подсветки (рис. 2).

2. Разбиение каждого видеосигнала на последовательность непересекающихся видеокладов (цифровых изображений).

3. Представление i -го цифрового изображения в цветовой модели RGB (т. е. получение 24-рядного рисунка).

4. Преобразование полученного цветного изображения в изображение с нулевым контрастом (изображение в градации серого) выполняется по формуле, которая отражает цветовое восприятие человека:

$$S_{n,m}(i) = \alpha R_{n,m}(i) + \beta G_{n,m}(i) + \gamma B_{n,m}(i),$$

где $R_{n,m}(i)$, $G_{n,m}(i)$, $B_{n,m}(i)$ — красный, зеленый, синий компоненты цветовой модели RGB; n, m — номер строки, столбца изображения; i — номер изображения (наблюдения); $\alpha = 0,30$, $\beta = 0,59$, $\gamma = 0,11$.

5. Минимаксное нормирование (индекс Н) изображения осуществляется по формуле

$$S_{n,m}^H(i) = \frac{S_{n,m}(i) - S^{\min}}{S^{\max} - S^{\min}},$$

где $S^{\max} = 255$, $S^{\min} = 0$ — максимальное и минимальное значения $S_{n,m}(i)$.

6. Адаптивная бинаризация (индекс Б) изображения ("1" — черный цвет пикселя, "0" — белый) осуществляется по следующему правилу:

$$S_{n,m}^B(i) = \begin{cases} 1, & \text{если } S_{n,m}^H(i) \leq \delta(i); \\ 0, & \text{если } S_{n,m}^H(i) > \delta(i), \end{cases}$$

где $\delta(i)$ — пороговый уровень, зависящий от яркости i -го изображения.

7. Элиминирование факторов, искажающих изображение фрагментов рельса, осуществляется следующим образом: если в заданной окрестности пикселя, у которого $S_{n,m}^B(i) = 1$, все другие пик-

сели имеют значение, равное нулю, то $S_{n,m}^B(i)$ этого пикселя принимается равным нулю. Размер окрестности очистки устанавливает квадрат с центром в пикселе, у которого цвет черный ("черный пиксель"). Размер, равный единице, соответствует квадрату 3×3 пикселей с центром в "черном пикселе", размер равный двум, — квадрату 5×5 и т. д. Если "черный пиксель" расположен близко к краю изображения, то часть окрестности, не имеющая пикселей, отсекается.

8. Получение признака путем суммирования бинарных кодов цвета пикселей в соответствии с выражениями (рис. 3) по строкам:

$$X_1(i) = \sum_{m=1}^{96} S_{1,m}^B(i), \dots,$$

$$X_{96}(i) = \sum_{m=1}^{96} S_{96,m}^B(i),$$

по столбцам:

$$X_{97}(i) = \sum_{n=1}^{96} S_{n,1}^B(i), \dots,$$

$$X_{192}(i) = \sum_{n=1}^{96} S_{n,96}^B(i).$$

9. Сшивка кадров, имеющих дефект, расположенный на границе соседних кадров.

10. Проверка условия присутствия дефекта на поверхности рельса. Если условие выполняется, то делается переход к операции 11, в противном случае — к условию 19.

11. Оценивание вероятности появления дефектов путем подачи на входы ИНС (многослойного перцептрона, в промежуточных и выходном слоях которого использованы нейроны сигмоидального типа с униполярной функцией активации) выделенного признака осуществляется по следующим формулам:

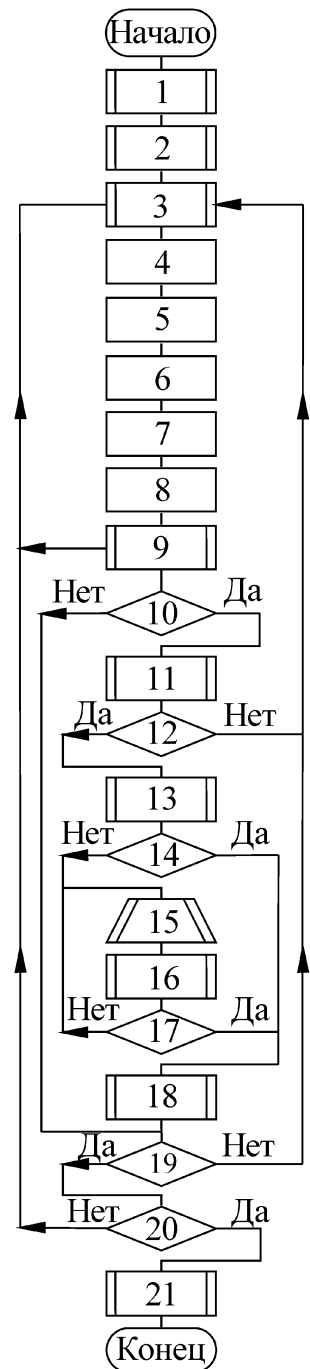


Рис. 1. Процедура распознавания поверхностных дефектов рельсов в потоке производства

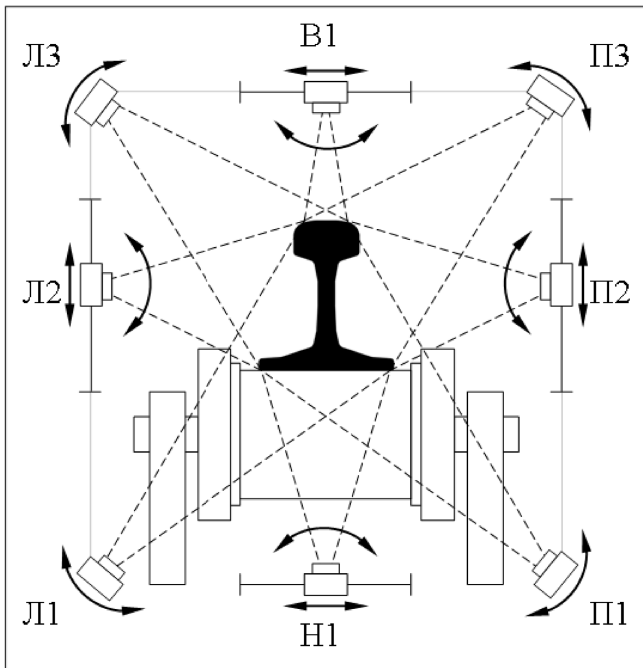


Рис. 2. Схема расположения камер, обеспечивающих "двух-" и "трехглазие": В1 и Н1, Л1 – Л3 и П1 – П3 – верхняя и нижняя, левые и правые видеокамеры

$$X_j^H(i) = \frac{X_j(i) - X_j^{\min}}{X_j^{\max} - X_j^{\min}};$$

$$Y_p^H(i) = \left[1 + \exp \left(- \sum_{m=0}^{100} w_{pm}^{(3)} \left[1 + \exp \left(- \sum_{k=0}^{100} w_{mk}^{(2)} \times \right. \right. \right. \right. \\ \left. \left. \left. \times \left[1 + \exp \left(- \sum_{j=0}^{192} w_{kj}^{(1)} X_j^H(i) \right) \right]^{-1} \right) \right]^{-1} \right]^{-1} ;$$

$$Y_p(i) = Y_p^{\min} + [Y_p^{\max} - Y_p^{\min}] Y_p^H(i),$$

где $X_j(i)$ – выделенный признак, $j = 1, \dots, 96, 97, \dots, 192$; $Y_p(i)$ – оценка вероятности возникновения p -го дефекта рельса, $p = 1, \dots, 8$ (выходной нейрон, у которого $p = 1$, отвечает за дефект "Плена № 1", если $p = 2$, то "Плена № 2", если $p = 3$, то "Плена № 3", если $p = 4$, то "Плена № 4", если $p = 5$, то "Плена № 5", если $p = 6$, то "Плена № 6", если $p = 7$, то "Раскатанная трещина", если $p = 8$, то "Раскатанный пузырь"); $X_j^H(i)$ и $Y_p^H(i)$ – нормированные значения $X_j(i)$ и $Y_p(i)$; X_j^{\max} , X_j^{\min} и Y_p^{\max} , Y_p^{\min} – максимальное и минимальное значения $X_j(i)$ и $Y_p(i)$, взятые из обучающей

выборки; $w_{kj}^{(1)}$, $w_{mk}^{(2)}$, $w_{pm}^{(3)}$ – весовые коэффициенты, оптимальные значения которых определяются в процессе обучения сети.

Если сеть не смогла однозначно установить тип дефекта, то его изображение последовательно поворачивается на разные углы. Для каждого полученного изображения вычисляется информативный признак, после чего он подается на входы сети. Параллельно запускается процедура сравнения информативных признаков этих изображений с информативными признаками эталонов типовых дефектов. Рассчитываются нормализованные коэффициенты корреляции между этими признаками. Наибольшее значение коэффициента корреляции подтверждает принадлежность наблюдения к одному из типов дефекта.

12. Проверка условия "Все ли варианты кадров рассмотрены?". Если условие выполняется, то переход к операции 13, в противном случае – возврат к операции 3.

13. Отнесение отдельного изображения к одному из типов дефекта с учетом дополнительной информации (например, маркировки рельса [12]).

14. Проверка условия "Тип дефекта установлен с заданной точностью?". Если условие выполняется, то переход к операции 18, в противном случае – переход к операции 15.

15. Визуальный осмотр оператором-контролером дефектного участка натурального рельса и его цифрового бинаризованного изображения.

16. Запуск ЭС распознавания дефектов, использующей продукционную модель представления знаний, как правило, применяемой в нестан-

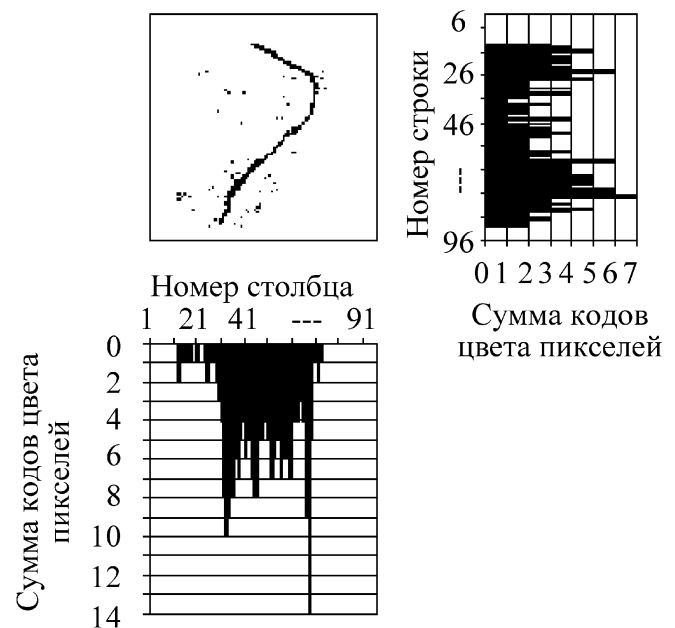


Рис. 3. Пример выделения информативного признака

дартных случаях, например, при идентификации редко встречаемых (например, один раз в год) нетиповых дефектов.

17. Проверка условий "Тип дефекта установлен с заданной точностью?", "Число повторений больше максимально возможного?". Если хотя бы одно условие выполняется, то переход к операции 18, в противном случае — возврат к операции 15.

18. Расчет координат и параметров дефекта.

19. Проверка условия "Все ли поверхности данного рельса обработаны?". Если условие выполняется, то переход к условию 20, в противном случае возврат к операции 3.

20. Проверка условия "Обработаны все изображения рельса?". Если условие выполняется, то переход к операции 21, в противном случае — возврат к операции 3.

21. Регистрация координат и параметров дефектов рельса.

Экспериментальная часть

Для моделирования предлагаемой процедуры первоначально авторами была сформирована обучающая выборка типовых дефектов рельсов (рис. 4), обучена ИНС, при этом наибольшую эффективность показал алгоритм обратного распространения ошибки, и создана ЭС на основе сочетания продукционного и балльного методов.

Создание ЭС включало выполнение следующих этапов:

- описание проблемной ситуации;
- извлечение знаний;
- структурирование и формализация знаний;
- программная реализация;
- моделирование и натурные испытания ЭС.

В качестве основных методов извлечения знаний были использованы: диалоги, дискуссии, интервью с экспертами (опытными операторами-контролерами), а также анализ текстов (ГОСТов, технологических инструкций, атласов). Эти методы позволяют осуществить перенос компетентности от эксперта к инженеру по знаниям.

На основе структурирования знаний получена таблица, увязывающая типы дефектов с их характерным графическим изображением, морфологическими свойствами (цветом, блеском, гладкостью, шероховатостью поверхности, геометрической формой), описанием микрошлифа, местом

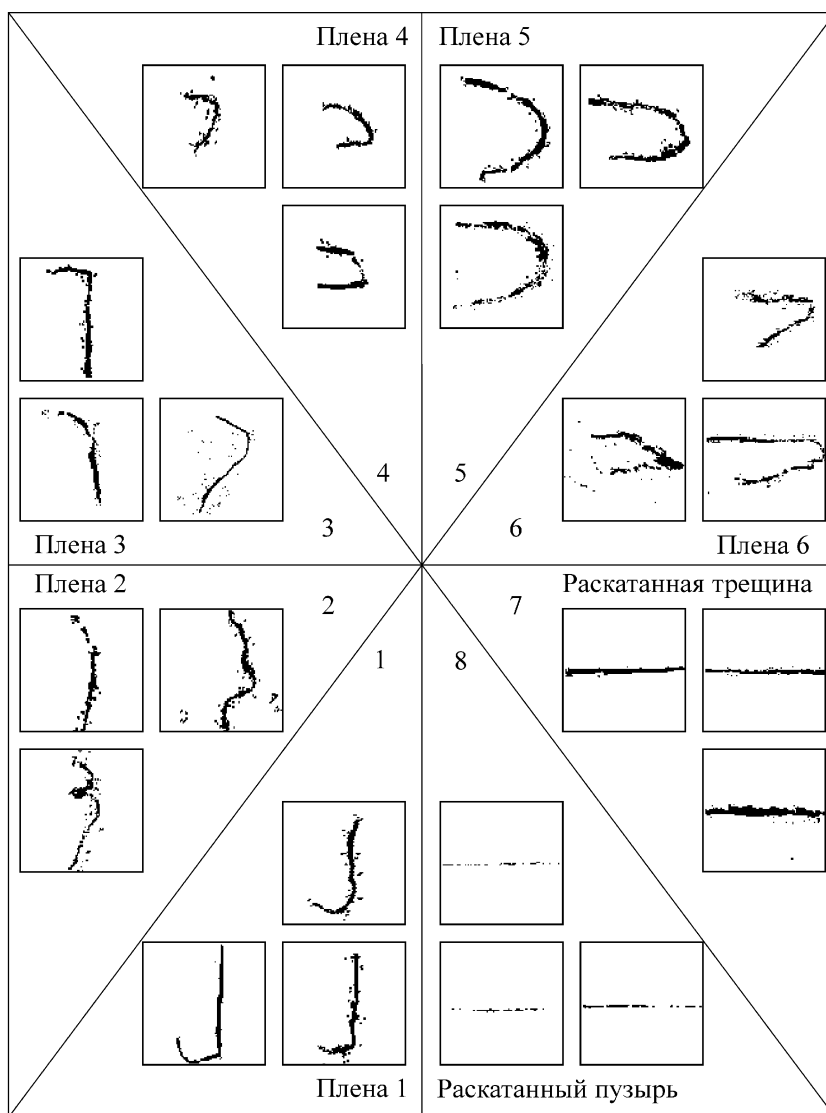


Рис. 4. Фрагмент обучающей выборки ИНС (восемь классов)

расположения (выкружка пера, перо, шейка, нижняя выкружка головки, боковая поверхность головки, верхняя выкружка головки, середина головки, верхняя выкружка головки, боковая поверхность головки, нижняя выкружка головки, шейка, перо, выкружка пера, подошва) и характером их расположения (единично или группами), протяженностью, глубиной и генетическими признаками (причины возникновения).

Для формализации знаний была использована продукционная модель, фрагмент которой имеет следующий вид: IF ("цвет" = "темно-серый", "блеск" = "отсутствует", "поверхность дефекта" = "гладкая", "геометрическая форма" = "отслоение металла языкообразной формы, соединенное с основным металлом одной стороной", "описание микрошлифа" = "в зоне дефекта наблюдается окалина, металл обезуглерожен и окислен", "элемент профиля рельса, на котором располагается

дефект" = "середина головки рельса", "характер расположения" = "единично", "протяженность" = "от 200 до 300 мм", "глубина" = "от 0,5 до 6,0 мм", "причины возникновения дефекта" = "заворот корочки на поверхности непрерывнолитой заготовки", ...), THEN ("суммарная балльная оценка (СБО) возникновения плены" = "48,73 баллов", "СБО возникновения раскатанного пузыря" = "15,14 баллов", "СБО возникновения волосовины" = "15,05 баллов", "СБО возникновения риски" = "12,04 баллов", "СБО возникновения раскатанной трещины" = "12,03 баллов", "СБО возникновения рванины" = "10,01 баллов", ...).

Для реализации ЭС-прототипа был выбран программный продукт Exsys CORVID, состоящий из среды разработки Development (логического (производственного) и командного блоков, блока "тип переменных") и среды исполнения Runtime.

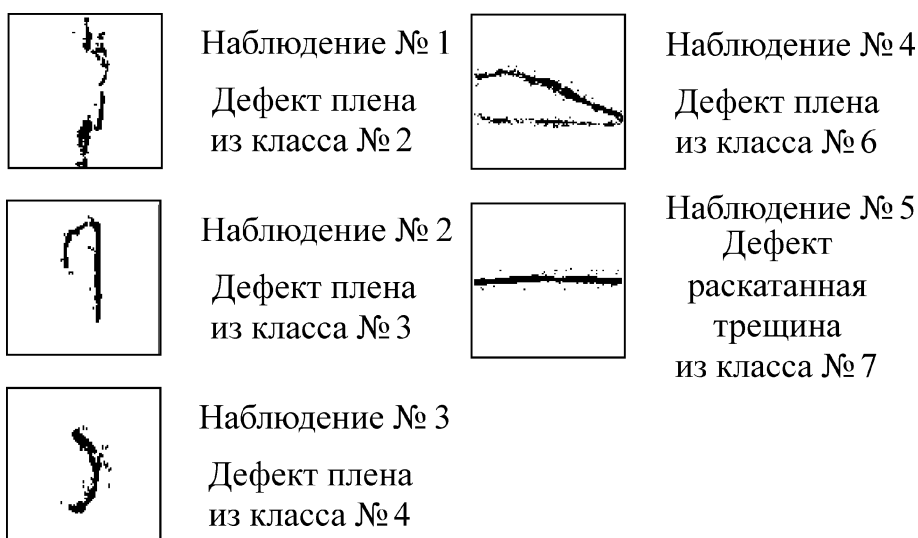


Рис. 5. Наблюдения из контрольной выборки ИНС



Рис. 6. Пример работы ИНС на контрольной выборке

Моделирование ЭС-прототипа осуществлялось с использованием натуральных объектов (рельсов) и их цифровых отображений (видеоинформации в виде последовательностей кадров). Для каждой ситуации оператор-контролер, визуально осматривая дефектный участок реального рельса и его цифровое изображение, очищенное путем бинаризации от фона и помех (см. рис. 3), отвечая на вопросы экспертной системы, получает суммарную балльную оценку возникновения конкретного дефекта. Объект, указанный на рис. 3, ЭС-прототип идентифицировал как плену.

После настройки ИНС и ЭС была сформирована контрольная выборка, фрагмент которой представлен на рис. 5, после бинаризации и очистки факторов, искажающих изображение. Используя контрольную выборку, ИНС смогла идентифицировать дефекты с эффективностью 90 %, за счет операции вращения изображения критерий качества распознавания составил 95 %, а при подключении ЭС-прототипа его значение увеличилось до 97 % (рис. 6).

Результаты

Осуществлена содержательная постановка задачи распознавания дефектов рельсов. Разработана новая процедура интеллектуального распознавания поверхностных дефектов рельсов в потоке производства. Предложена новая схема установки, обеспечивающая полный контроль всех поверхностей рельса в режиме реального времени.

Результаты моделирования описанной системы по ситуациям с использованием натуральных видеок кадров, полученных в рельсобалочном цехе ОАО "НКМК", показали возможность достижения 90–97 % эффективности. Преобразование построенного прототипа ЭС в промышленный ее вариант позволит достигнуть практически 100 % эффективности за счет пополнения базы знаний, настройки технических узлов, модернизации ИНС.

Список литературы

1. **Методы** компьютерной обработки изображений / Под ред. В. А. Сойфера. М.: ФИЗМАТЛИТ, 2003. 784 с.
2. **Цыпкин Я. З.** Адаптация и обучение в автоматических системах. М.: Наука, 1968. 216 с.
3. **Горелик А. Л., Скрипкин В. А.** Методы распознавания: Учебн. пос. для вузов. М.: Высш. школа, 2004. 261 с.
4. **Люгер Дж. Ф.** Искусственный интеллект: стратегии и методы решения сложных проблем. М.: Издат. дом "Вильямс", 2003. 864 с.
5. **Фу К.** Последовательные методы в распознавании образов и обучении машин. М.: ФИЗМАТЛИТ, 1969. 328 с.
6. **Дуда Р., Харт П.** Распознавание образов и анализ сцен. М.: Мир, 1970. 512 с.

7. **Дефекты** и качество рельсовой стали: Справ. изд. / В. В. Павлов, М. В. Темлянец, Л. В. Корнева и др. М.: Теплотехник, 2006. 218 с.
8. **Дефекты** стальных слитков и проката: Справ. изд. / В. В. Правосудович, В. П. Сокуренок, В. Н. Данченко и др. М.: Интермет Инжиниринг, 2006. 384 с.
9. **Дефекты** в металлах; Справочник-атлас / А. А. Ежов, Л. П. Герасимова. М.: Русский университет, 2002. 360 с.
10. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных системы. СПб.: Питер, 2001. 384 с.
11. **Awad E.** Building knowledge automation expert systems with Exsys CORVID. Charlottesville: University of Virginia, 2005. 265 p.
12. **Автоматическая** идентификация рельсов в потоке производства / С. М. Кулаков, В. Б. Трофимов, Н. Ф. Бондарь, С. В. Чабан // Математические методы в технике и технологиях — ММТТ-19: Тр. XIX Междунар. научн. конф. Воронеж: ВГТА, 2006. С. 78—81.

УДК 004:656

Р. М. Алгулиев, чл.-корр. НАНА, проф.,
Р. С. Абдуллаев,
Институт информационных технологий НАНА,
г. Баку

Анализ способов сокращения транспортных нагрузок на городскую инфраструктуру

Дан обзор путей разрешения проблем автомобильных пробок. Описываются способы сокращения автомобильных пробок и опыта развитых стран для решения этой проблемы.

Проблема исследования

Общей чертой крупных городов являются транспортные дорожные пробки, которые в густонаселенных городах приводят, с одной стороны, к ненужным простоям и характеризуются временными и экономическими потерями, а с другой — ведут к ухудшению экологической обстановки. А это, в свою очередь, не позволяет обеспечить предоставление различных услуг: услуг "скорой помощи", распределение товаров, транспортировку опасных грузов, вывоз различных видов мусора, курьерскую службу и т. д. [1—4].

Ведущие специалисты различных отраслей ищут пути решения этой актуальной проблемы. Вследствие проведенных работ на дорогах многих городов удалось частично уменьшить транспортные нагрузки, но до разрешения этой проблемы пока очень далеко.

Данная статья посвящена анализу путей решения этой непростой проблемы.

Обзор состояния проблемы борьбы с автомобильными пробками

В последнее время широко используется автомобильная навигация. Все больше и больше ком-

паний отдадут предпочтение системам NAVISTAR-GPS, или просто GPS. Такие системы очень распространены в морском судоходстве и авиации. В последнее время приемники GPS используются и для автомобильной навигации [4—15].

Система GPS способна определять координаты местонахождения мобильного объекта (например, высоту его от уровня моря, направление и скорость движения). Следует отметить, что в экологических, метрологических и охранных службах эта система себя оправдала. В настоящее время уже получили широкое распространение точные и недорогие навигационные приемники, которыми могут пользоваться рядовые граждане.

Так, в системе GPS каждый водитель для передачи информации в диспетчерский пункт снабжается приемниками GPS и радиотехническими средствами. При этом на мониторе экрана диспетчера с помощью специальных программных приложений формируется электронная карта обслуживаемой территории. После поступления по радиоканалу суммарной информации о координатах мобильного объекта и его скорости на электронной карте уточняется местонахождение мобильного объекта. Кроме информации о координатах мобильного объекта на электронные карты с помощью радиоканала передается различная информация технического и экономического характера.

Для достижения поставленной цели надо разработать электронную карту зоны обслуживания. В ней указывают местоположение координат объектов государственной важности, а также место дорожных указателей и светофоров [17]. Использование электронной карты и радиотелефонного канала резко повышает вероятность выбора оптимального маршрута движения. В свою очередь, при оптимальном выборе направления маршрута на электронной карте число пробок на дорогах удается резко уменьшить [10—12].

Одним из способов уменьшения числа пробок на дорогах является дистанционное управление светофорами. Мировой опыт показывает, что управление дорожным движением с использованием интеллектуальных сетей с конечными нагрузками в виде светофоров — очень эффективный способ по динамичному управлению транспортными потоками. Высокотехнологичные светофоры на базе современных интегральных микросхем могут точно определить скорость движения на дорогах и дорожных узлах. Поэтому целесообразно снабдить их специальными средствами, позволяющими управлять интенсивностью транспортных потоков. Таким образом, полученная информация с использованием локальной сети может быть применена при распределении нагрузки на участках между светофорами. Интенсивность загорания красного и зеленого света на светофорах позволяет резко уменьшить число простоев на дорогах. Для этого необходимо управление светофорными группами с единого центра, а также поступление информации к установленным на автомобильных бортах компьютерам и навигационным системам. Другими словами, все водители должны оповещаться в реальном времени информацией в виде изображения, голоса или текста [18—20].

Проведенные исследования показывают, что транспортные пробки характеризуются огромными потерями времени для людей, находящихся на дорогах. Как выход из положения, во многих городах в точках пересечения дорог монтируют специальные следящие камеры и системы управления дорожным движением. Работа этих систем основана на получении сигналов от автоматически управляемых дорожных светофоров. Интенсивность загорания красного и зеленого света светофоров автоматически изменяется в зависимости от числа пробок, в результате чего текущее состояние дорог с помощью электронных табло, установленных в различных местах города, позволяет выводить эту информацию на мониторы для всех водителей.

Уже создана экспериментальная модель такой сети под названием *Neuromonet*, где транспортная система представлена в виде нейронной сети, а автомобили и пешеходы — в виде нервных импульсов. В крупных городах, таких как Токио, Лос-Анджелес, Нью-Йорк, Чикаго, Хьюстон, следящие системы за дорожным движением уже нашли достойное применение. Есть такое мнение, что использование интеллектуальной сети является самым эффективным способом увеличения пропускной способности городских автомобильных дорог [20—24].

Опыт крупных городов мира показывает, что вывод многих предприятий за черту города явля-

ется одним из самых эффективных способов уменьшения транспортных пробок. Безусловно, такая практика позволяет резко уменьшить число транспортных пробок, но полностью решить эту проблему не удастся. Дело в том, что эти мероприятия требуют времени и больших вложений инвестиций.

Одним из факторов, вызывающих возникновение автомобильных пробок, является низкое качество дорожного покрытия автомобильных дорог. Резкое возрастание из года в год транспортных средств увеличивает интенсивность движения по автомобильным дорогам, что отрицательно сказывается на качестве дорожного покрытия. В совокупности эти факторы ведут к ненужным простоям транспортных средств на дорогах. Поэтому нужны периодические мероприятия по замене и ремонту дорог. В настоящее время в больших городах ведутся работы по построению новых магистральных дорог, дорожных узлов, многоярусных мостов, туннелей, подземных переходов и многоэтажных гаражей. В общем, все эти мероприятия необходимы, однако они не могут полностью решить проблему пробок.

Одним из скрытых источников создания пробок является неточная информационная обеспеченность при огромном количестве информационных ресурсов. Нахождение нужной информации, по сути, тоже является одной из проблемных задач. Необходимо создание информационных ресурсов общего пользования. Все страны должны постепенно переходить в информационное общество, которое предполагает электронизацию всех сторон человеческой жизни. В настоящее время во всем мире идут работы по созданию электронного пространства. Можно привести примеры по электронизации общества разных стран. Так, в гонконгском варианте огромное число людей использует 130 услуг более 40 государственных организаций. Например, используя веб-портал, можно выплачивать госналоги, участвовать в процессе бракосочетания, подавать заявление для получения разрешения, заниматься коммерческими делами, участвовать в спортивных мероприятиях и проектных работах, заниматься меценатством и т. д. Причем для выполнения перечисленных услуг граждане не используют транспортные средства, в результате чего транспортная нагрузка на городских дорогах значительно уменьшается.

Для решения этой проблемы власти Токио приступили к реализации масштабного проекта внедрения системы определения местонахождения, который предусматривает размещение в черте города около 10 тыс. RFID-меток и других радиомаяков. Метки помогут людям, вооруженным соответствующими терминальными устройства-

ми, получать информацию о местоположении интересующих их объектов. В процессе работы система сопоставляет уникальные коды, посылаемые маяками, с данными, которые хранятся на сервере, подключенном к Internet. Терминалы автоматически связываются с сервером через беспроводное соединение локальной сети и запрашивают информацию, соответствующую выбранному маяку. Система сможет предоставить пользователям средства базовой навигации, а также информацию о торговых точках в указанном районе. При приближении пользователя к RFID-метке, закрепленной, скажем, на уличном фонаре, на его терминал поступит информация о местоположении. Сможет система и указать маршрут до ближайшей станции метро. При попадании в зону действия радиомаяка, расположенного перед торговым заведением, человек получит возможность ознакомиться со специальными предложениями продавцов, а также с меню, которое предлагается в ресторане, находящемся в этом же здании. Имея же в своем распоряжении систему подобного рода, достаточно нажать на кнопку, и людям не составит труда ориентироваться на местности [25].

Для решения автомобильных пробок, как было отмечено, необходима организация предоставления услуг в общественных местах. Наиболее перспективным решением является организация и использование сенсорных киосков [26]. Такие киоски можно использовать в библиотеках, в государственных органах власти, территориальных подразделениях и федеральных учреждениях, в финансовых учреждениях и банках, в центре занятости. Мировой опыт показывает, что использование сенсорных киосков помогает людям найти нужную информацию вовремя, и не используя транспортных средств.

Выводы

Анализ состояния проблемы борьбы с автомобильными пробками позволяет сделать следующие выводы.

1. Для достижения поставленной задачи необходимо создание информационных ресурсов общего пользования. Поэтому все страны должны постепенно переходить к информационному обществу, которое ведет к электронизации всех сторон человеческой жизни.

2. Необходимо использовать электронный документооборот и технологию электронной подписи. При электронном документообороте уменьшается время обработки документов между исполнителями, что повышает эффективность выполняемых работ с помощью электронной подписи. Проведение мероприятий по электро-

низации документооборота также позволит уменьшить число транспортных услуг, и, как следствие, уменьшить транспортную нагрузку на городских автодорогах, что в свою очередь уменьшит число искусственных пробок. Проведенные мероприятия повысят качество обслуживания клиентов различных категорий с доставкой нужных товаров в заранее предусмотренные интервалы времени.

3. Для достижения поставленной цели необходимо оптимальное управление наземными транспортными средствами, а это предполагает использование специализированной централизованной автоматизированной управляющей системы. Такая система должна быть построена на базе современных информационных технологий и выполнять следующие основные функции:

- оперативное слежение за маршрутами городского движения;
- перераспределение автомобилей на напряженных маршрутах движения;
- прокладка и выбор оптимальных путей следования;
- немедленное реагирование на дорожно-транспортные происшествия.

Для управления транспортными потоками в соответствии с указанными функциями следует разработать электронно-цифровую карту зоны обслуживания и организовать диспетчерский пункт, построенный на базе современных технологий и работающий в составе корпоративной IP-сети.

4. Применение технологии интеллектуальных сетей и разработка электронной карты городов позволит оптимизировать доставку товаров и виды обслуживания после построения новых объектов в городах. По сути, эти мероприятия являются эффективным способом уменьшения пробок, транспортных нагрузок.

5. Для решения этой проблемы необходимо широко применять продукты современных информационных технологий.

Следует отметить, что выполнение этих мероприятий, хотя и потребует первоначального вложения инвестиций, но принесет безусловную экономическую выгоду и отдельным гражданам, и государственным учреждениям.

Список литературы

1. <http://www.uran.donetsk.ua>
2. www.hardwareportal.ru
3. <http://www.gpsinfo.ru>
4. <http://gps.guides-and-gear.com>
5. <http://www.navcen.uscg.gov>
6. <http://www.satellite.srd.mtuci.ru>
7. Николаев В. Местоопределение абонентов в сетях сотовой связи // Специальная техника. 2001. № 5.
8. Шебшаевич В. С., Дмитриев П. П., Иванцевич Н. В. и др. Сетевые спутниковые радионавигационные системы. М.: Радио и связь, 1993.

9. Волков Н. М., Иванов Н. Е., Салищев В. А., Тюбалин В. В. Глобальная навигационная спутниковая система "ГЛОНАСС" // Успехи современной радиоэлектроники. 1997. № 1.
 10. <http://www.gisa.ru>
 11. <http://www.geomatica.kiev.ua>
 12. Безрученко С. Системы диспетчерского управления и контроля транспортных средств на базе технологий спутниковой навигации // Информост. Средства связи. 1999. № 2.
 13. <http://www.tscc.ru>
 14. <http://www.avlinfosys.com>
 15. <http://kiev-security.org.ua>
 16. <http://www.ccc.ru>

17. Белов М. В. Топологическое разбиение цифровой карты для визуализации на мобильном терминальном оборудовании распределенных ГИС // Телекоммуникации. 2005. № 9.
 18. <http://www.avtomarket.ru>
 19. <http://www.upmark.ru>
 20. <http://www.ofib.org>
 21. <http://science.compulenta.ru>
 22. <http://express.kirov.ru>
 23. <http://www.terralab.ru>
 24. <http://www.iOne.ru>
 25. Computerworld Россия. № 3 (548), 2007.
 26. <http://www.touch.ru>



ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

УДК 004.891.2

Л. Е. Муханов,
ЗАО МЦСТ

Система обнаружения мошенничества в области платежных карт

В последнее время наблюдается явная тенденция роста мошенничества с использованием платежных карт, несмотря на переход на карты нового типа с микрочипом и постоянно совершенствующаяся система защиты. Поэтому в настоящее время есть потребность в системе предотвращения мошенничества, большинство функций принятия решений которой было бы автоматизировано. В данной статье приводится описание разработанной системы и одной из базовых моделей обнаружения мошенничества. В заключение приводятся результаты оценочного тестирования и делаются выводы о преимуществах и недостатках использования данной модели.

Введение

Платежная карта как форма оплаты постепенно вытесняет традиционные денежные носители. Но в последнее время наблюдается явная тенденция роста мошенничества с использованием платежных карт несмотря на процесс интеграции карт нового поколения с микрочипом (EMV) и постоянно совершенствующаяся системы защиты. Финансовые организации, работающие с платежными картами, стремятся внедрять различные системы предотвращения мошенничества. Большинство систем базируются на использовании правил, которые формируются на основе знаний соответствующих экспертов из банковской области. Но в результате постоянно увеличивающегося

потока транзакций и появления новых видов мошенничества становится достаточно сложно быстро выявлять подобные случаи и создавать соответствующие правила. Поэтому в настоящий момент существует потребность в системе, которая позволяла бы среди всего потока транзакций, обрабатываемых процессинговой системой банка, достаточно быстро выявлять новые виды мошенничества и предотвращать его.

По данным аналитической компании "Frost&Sullivan", в 2005 г. совокупные потери финансовых организаций от карточного мошенничества во всем мире составили 7,9 млрд долл. США, а к 2009 г. этот показатель должен вырасти до 15,5 млрд долл. США [1].

В статье приведено общее описание разработанной системы обнаружения мошенничества, а также одной из базовых моделей выявления мошенничества и результатов ее оценочного тестирования.

Механизмы выявления мошенничества

Большинство существующих механизмов выявления мошенничества базируются на использовании эвристических правил. Эти правила позволяют на основе параметров анализируемой транзакции относить ее к определенной группе риска. Также на их основе может проводиться анализ целой последовательности транзакций, что позволяет лучше выявлять случаи кражи реквизитов платежных карт в одной из стран повышенного риска, связанного с мошенничеством. Но в результате постоянно увеличивающегося потока транзакций подобный механизм требует все большего числа специалистов для своевременного выявления новых видов мошенничества и введения соответствующих правил. Также необходимо отметить, что из-за роста числа правил процесс сопоставления транзакций требует больше времени и вычислительных ресурсов. В качестве альтернативного ре-

шения могут использоваться и другие подходы, например, нейронные сети или сети Байеса [2]. Но недостаток использования подобных моделей связан с полной автономностью их функционирования (т. е. процесс обучения и классификации), что может привести к ошибке работы модели, исправить которую будет достаточно сложно. Поэтому необходима модель, которая, с одной стороны, устраняет недостатки, связанные с использованием правил, а с другой стороны, позволяет специалистам банка проводить различные настройки, связанные с процессом классификации транзакций.

В качестве базового механизма для выявления мошенничества в разработанной системе используется подобная модель. Данная модель позволяет на базе соответствующей статистической информации определить, насколько классифицируемая транзакция характерна для легальных или мошеннических транзакций. В качестве статистической информации используются области кластеризации значений параметров транзакций. Приведем пример: если в выборке, предназначенной для обучения модели, значения одного из полей транзакций, осуществленных с помощью одной и той же платежной карты, распределены определенным образом (табл. 1), то будут выделены определенные центры кластеризации и соответствующие отклонения (табл. 2).

Таким образом, если значение поля классифицируемой транзакции равно 760, то оно признается характерным для данного класса транзакций. Получение центров кластеризации происходит следующим образом.

Для каждого из параметров транзакции находится максимальная разница ($DIFF_{max}$) значений

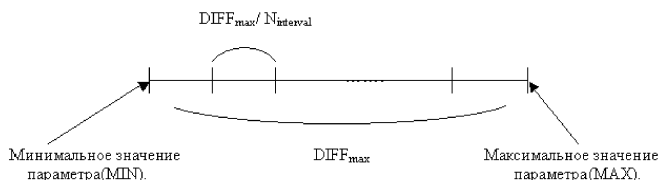


Рис. 1. Разбиение максимальной разницы значений на интервалы

этого параметра, зарегистрированных в процессе сбора статистики. Далее эта разница ($DIFF_{max}$) разбивается на $N_{interval}$ отрезков. $N_{interval}$ вычисляется как двоичный логарифм числа значений параметра N_{points} . Вообще говоря, число отрезков ($N_{interval}$), на которое разбивается $DIFF_{max}$, может выбираться произвольным образом. В данном случае $N_{interval}$ вычисляется исходя из того, что при увеличении значений параметра N_{points} в 2 раза число используемых отрезков ($N_{interval}$) увеличивается на единицу (рис. 1).

Далее для каждого из полученных отрезков вычисляется среднее значение и максимальное отклонение от этого среднего значений параметра, попавших в этот интервал. Таким образом, получается $N_{interval}$ центров и соответствующих отклонений, которые и являются описанием распределения значений соответствующего поля транзакции. Таким образом, все $N_{interval}$ центров (за исключением центров, относящихся к отрезкам, в которые не попало ни одно из значений параметра) отклонений используются в качестве статистической информации для осуществления классификации транзакций.

Данная информация набирается на основе проверенных сотрудниками банка транзакциях во время обучения системы отдельно для легальных транзакций, осуществленных с помощью соответствующей платежной карты, для всех легальных транзакций и для всех мошеннических транзакций. Классифицируемая транзакция сначала сопоставляется с областями кластеризации, полученными из легальных транзакций, которые были осуществлены с помощью данной платежной карты, и всех легальных транзакций. В случае, если обнаруживается, что классифицируемая транзакция не принадлежит ни одному из этих классов, то она сопоставляется со множеством мошеннических транзакций. В случае, если транзакция классифицируется данной моделью как мошенническая, то она признается подозрительной.

Сам процесс классификации в данной модели сводится к следующему. Значение каждого из параметров классифицируемой транзакции, сопоставляется с соответствующими центрами кластеризации, полученными в процессе сбора статистики. Если данное значение входит (т. е. отклонение значения от центра кластеризации не

Таблица 1

Представление значений одного из параметров

Номер значения	Значение параметра
1	1200
2	1220
3	1260
4	1270
5	720
6	730
7	780
8	800

Таблица 2

Представление центров кластеризации и отклонений для одного из параметров

Номер кластера	Значение кластера	Максимальное отклонение значений внутри этого кластера от его центра
1	757,5	42,5
2	1237,5	37,5

превышает допустимого значения) хотя бы в один из центров кластеризации для соответствующего параметра, то в качестве результата классификации ($Class_1$) возвращается 1 (т. е. значение параметра характерно для данного класса). Если же значение не входит ни в один из центров кластеризации, то в качестве результата классификации ($Class_1$) возвращается -1 (т. е. значение параметра нехарактерно). Окончательный результат классификации транзакции представляет собой линейную комбинацию результатов классификации каждого из параметров транзакции [3]:

$$Result = w_1 Class_1 + w_2 Class_2 + \dots + w_n Class_n,$$

где n — число параметров; w_1, \dots, w_n — соответствующие коэффициенты (в дальнейшем весовые коэффициенты), причем $w_i = 1/n$ ($i = 1, \dots, n$).

Таким образом, если $Result$ больше 0, то транзакция признается характерной для соответствующего класса, если же $Result$ меньше 0, то транзакция не является характерной. $Result$ по модулю — это числовое значение, отражающее точность результата классификации.

Весовые коэффициенты могут варьироваться в зависимости от значимости результата классификации каждого из параметров транзакции. Если результат классификации параметра не должен отражаться на окончательном результате классификации всей транзакции или параметр не имеет информационной нагрузки, то весовой коэффициент может быть уменьшен [3].

Для автоматизированного определения весовых коэффициентов все данные (в данной реализации все легальные транзакции), предназначенные для набора статистики, делятся на две части. Первая часть используется непосредственно для осуществления сбора статистической информации о параметрах транзакций. Вторая часть используется непосредственно для получения весовых коэффициентов. Определение этих коэффициентов происходит путем осуществления классификации транзакций, содержащихся во второй части, на основе статистических данных о параметрах транзакций, полученных из первой части. Если в процессе классификации для любой транзакции оказывается, что значение какого-либо параметра нехарактерно (т. е. оно не входит в один из центров кластеризации), то его весовой коэффициент в данной реализации уменьшается в 2 раза:

$$w'_i = w_i/2.$$

Если в процессе классификации транзакции был изменен какой-либо из весовых коэффициентов, то возникает необходимость провести нормализацию этих коэффициентов:

$$w'_i = w_i / \left(\sum_n w_j \right).$$

Экземпляры мошеннических транзакций также могут быть использованы для получения весовых коэффициентов, для этого необходимо провести их классификацию на основе набранной статистической информации о параметрах легальных транзакций. В случае, когда значение параметра мошеннической транзакции не входит ни в один из центров кластеризации значений параметров легальных транзакций, то весовой коэффициент соответствующего параметра увеличивается (в данной реализации в два раза).

Общее описание системы

В разработанной системе для обнаружения мошеннических транзакций используются два отдельных модуля: *FDS_ONLINEP* и *FDS_OFFLINEP* (рис. 2). Модуль *FDS_ONLINEP* используется для выявления мошенничества среди транзакций, поступающих от фронтальной компоненты процессинговой системы при их авторизации. В данном модуле в случае обнаружения подозрительной транзакции в процессинговую систему отправляется соответствующее уведомление, на основе которого и будет принято решение об авторизации транзакции.

Для выявления мошенничества с помощью моделей, которые не могут быть задействованы в модуле *FDS_ONLINEP* из-за временных ограничений, накладываемых на процесс авторизации транзакций, используется модуль *FDS_OFFLINEP*. Этот модуль позволяет выявлять мошенничество среди осуществленных транзакций, которые уже были проверены в модуле *FDS_ONLINEP* или поступили из какого-либо внешнего источника. Их хранение осуществляется в соответствующей базе данных (*FDS DATA WAREHOUSE*). В ней также хранится статистическая информация, набранная во время обучения системы, параметры настройки системы, результаты классификации транзакций различными моделями.

Модуль *FDS_ALERTP* предназначен для предупреждения владельца платежной карты средствами *SMS*-сообщений или электронной почты в случае обнаружения подозрительной транзакции в модуле *FDS_ONLINEP*.

Для обучения системы используется модуль *FDS_BUILDSTATP*, который позволяет осуществлять и согласовывать процесс обучения используемых моделей выявления мошенничества с другими модулями системы (*FDS_ONLINEP* и *FDS_OFFLINEP*).

Для выявления мошеннических транзакций в системе используются следующие модели:

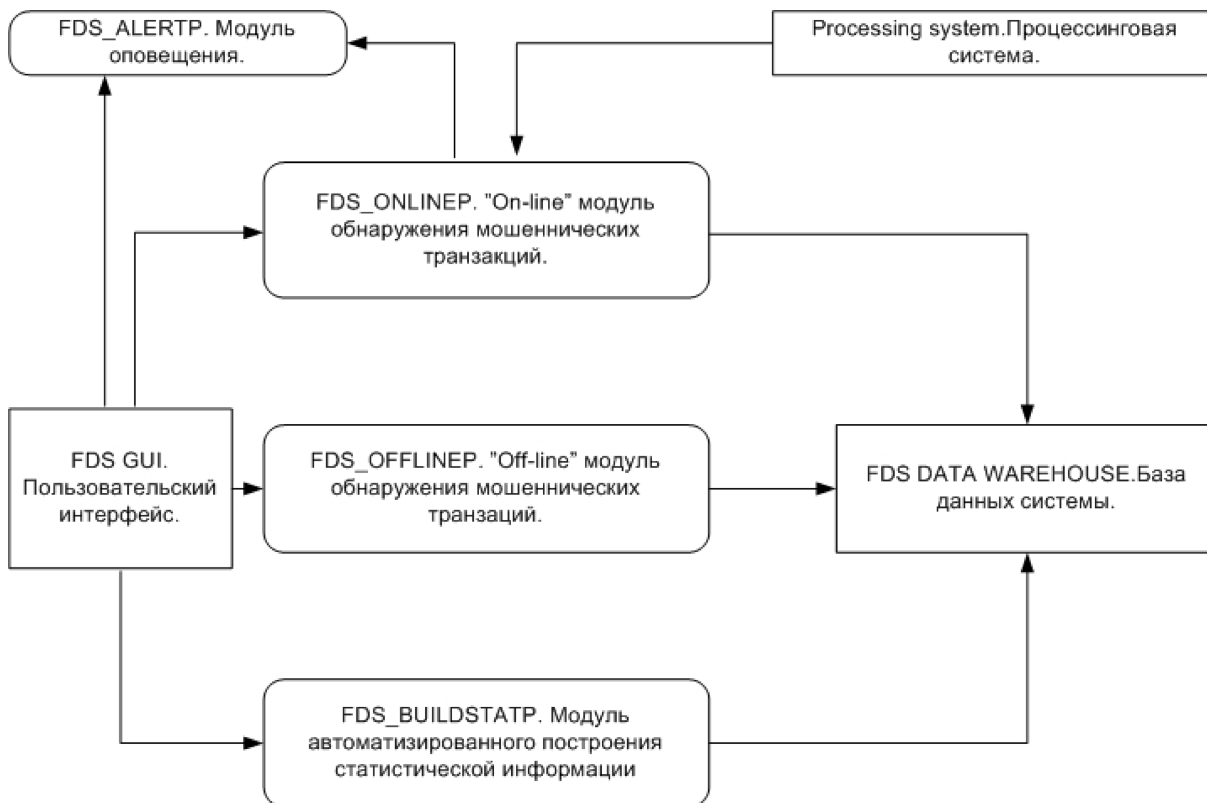


Рис. 2. Общая архитектура системы обнаружения мошеннических транзакций

- модель, базирующаяся на использовании областей кластеризации значений параметров транзакций. Эта модель позволяет проводить анализ легальных (и мошеннических) транзакций (их параметров) и находить области кластеризации значений, по которым определяется, насколько классифицируемая транзакция характерна для легальных или мошеннических транзакций;
- сети Байеса. Эта модель позволяет получить вероятностную оценку того, насколько классифицируемая транзакция характерна для мошеннических или легальных транзакций;
- модель, базирующаяся на использовании географического места осуществлений транзакций. Например, в случае, когда транзакции были осуществлены из разных географических точек, а разница во времени между их осуществлением подозрительно небольшая, то транзакция признается мошеннической;
- модель, базирующаяся на эвристических правилах. Она позволяет выявлять подозрительные транзакции или их последовательности на основе правил, которые создаются экспертами банка;
- модель, базирующаяся на использовании ограничений (лимитов). Данная модель позволяет выявлять подозрительные транзакции на основе ограничений (например, число транзакций

в месяц с карты, сумма транзакций в месяц и др.), которые создаются экспертами банка.

Результаты оценочного тестирования

Для оценочного тестирования была создана искусственная выборка транзакций, часть которых соответствует использованию платежной карты обычным пользователем, а часть соответствует типичным мошенническим транзакциям. Данная выборка в основном предназначена для оценки того, как разработанная модель ведет себя в простых и наиболее встречающихся случаях использования карты. Процесс тестирования моделей проводился на основе 56 транзакций, предназначенных для обучения системы, и 11 транзакций, предназначенных для классификации. Все транзакции были осуществлены с одной и той же карты. Среди транзакций, предназначенных для обучения, 52 экземпляра соответствуют классу легальных транзакций, а 4 — классу мошеннических транзакций. Легальные транзакции были осуществлены с использованием 6 различных собственных банкоматов банка, 12 внешних финансовых институтов (т. е. через платежную систему). Транзакции проводились в разное время и в разных странах. Мошеннические транзакции были осуществлены из определенного финансового института определенной страны и в определенное время. Сре-

ди транзакций, предназначенных для классификации, девять транзакций содержат параметры, характерные для легальных транзакций, а две соответствуют классу мошеннических транзакций.

Результаты оценочного тестирования приведены на рис. 3 (см. третью сторону обложки). В данном тестировании 9 из 11 транзакций были классифицированы как легальные с большой степенью вероятности. Причем транзакции под номерами 7 и 8 имеют один нехарактерный параметр (время осуществления транзакции) при сопоставлении с классом легальных транзакций. Результат сопоставления транзакций под номерами 10 и 11 с классом легальных транзакций оказался отрицательным, поэтому данные транзакции были сопоставлены с классом мошеннических транзакций и был получен положительный результат.

Из полученных результатов можно сделать вывод о том, что модель ведет себя адекватно ситуации в данном тестировании. Основное преимущество данной модели заключается в том, что она позволяет выводить список всех нехарактерных параметров как при сопоставлении с легальными,

так и при сопоставлении с мошенническими транзакциями (т. е. удобно использовать при мониторинге и анализе транзакций), а также проводить настройку системы за счет изменения весовых коэффициентов. Дальнейшее усовершенствование данной модели возможно за счет использования существующих зависимостей между параметрами транзакции, что значительно влияет на точность результата классификации [4].

Список литературы

1. **EMV Will Push Card Fraud To Next Weakest Link** [Электронный ресурс] // The payment news and resource center (Epaynews): [web-сайт]. 09.11.2005. [http://www.epaynews.com/index.cgi?survey = &ref = browse&f = view&id = 113413610062215212 &block \(02.02.2007\)](http://www.epaynews.com/index.cgi?survey = &ref = browse&f = view&id = 113413610062215212 &block (02.02.2007)).
2. **Муханов Л. Е.** Адаптация модели сетей Байеса для обнаружения мошенничества с платежными картами // Информационные технологии. 2008. № 2. С. 74—78.
3. **Friedman J. H., Hastie T., Tibshirani R.** Additive logistic regression: a statistical view of boosting // *Annalyse of Statistics*. 2000. P. 337—407.
4. **Lam Wai, Bacchus Fahiem.** Learning Bayesian belief networks: An approach based on the MDL principle // *Computational Intelligence*. 1994. Vol. 10. № 4. P. 269—293.

УДК 004:658.011.56

А. В. Дрогин,

"МАТИ"-РГТУ имени К. Э. Циолковского

Разработка научно-методического и программного обеспечения подсистемы бюджетирования АСУП

Обсуждается разработка программного обеспечения подсистемы бюджетирования, входящей в состав автоматизированной системы управления предприятием. Представленная разработка объединяет в единый инструмент оперативного управления производственным циклом, планирования деятельности предприятия и анализа результатов две независимые подсистемы: управления производственным циклом и финансового планирования.

Актуальность проблемы

Планирование — одна из основополагающих функций управления предприятием, поскольку именно в задачи планирования входит определение оперативных и стратегических целей деятельности предприятия и методов их достижения. Однако планирование не может осуществляться в отрыве от действительности — основанием для

планов всегда являются достигнутые ранее результаты. Кроме того, даже составленные предельно тщательно планы не всегда выполняются. В таком случае необходим анализ причин срыва плана и их учет при последующем планировании. Эффективно решать перечисленные задачи может *бюджетирование* [1, 2].

Тем не менее, при всех достоинствах бюджетирование является сложным инструментом управления предприятием и требует аккуратного и квалифицированного подхода при использовании. В частности, бюджетирование запасов готовой продукции требует точного прогнозирования многих параметров производственного процесса: спроса на продукцию, времени поставки материалов, доступности производственных мощностей и т. д. Описанная в статье разработка позволила фактически устранить промежуточный этап сбора разрозненных данных, которые должны быть объединены в процессе финансового планирования.

Средства автоматизации бюджетирования представлены на российском рынке систем управления предприятиями достаточно широко. Особо следует отметить специализированные программные продукты компании "Инталев" [3], предлагающие наиболее широкий выбор инструментов бюджетного управления. Однако специализированные средства бюджетирования, как правило, слабо интегрированы в корпоративные информационные системы,

прочие же программные продукты, реализованные на общей с рассматриваемой системой платформе, предлагают недостаточный для решения поставленных задач функционал и не отражают специфики производственного предприятия.

Удобные в использовании средства бюджетирования представлены в программных продуктах 1С компании. Помимо удобства преимущество продуктов 1С заключается в глубокой интеграции с широко распространенной в России системой 1С: Предприятие. Однако продукты 1С имеют значительные ограничения по масштабируемости: производительность системы приемлема, если число пользовательских сессий не превышает 20 (для версии ПО 7.7) [4, 5].

Место бюджетной подсистемы в общей структуре АСУП

Автоматизированная система управления предприятием Microsoft Navision предназначена для управления предприятиями малого и среднего бизнеса. Система позволяет автоматизировать различные направления деятельности предприятия: финансовый учет, закупки, продажи, складскую логистику, учет основных средств, планирование производства и т. д. В число автоматизируемых направлений попадает и бюджетирование. Однако подсистема бюджетирования Navision не удовлетворяет потребностям планирования, поскольку реализует крайне упрощенную схему бюджетного процесса. По принятой разработчиками АСУП схеме бюджет представляет собой запланированные обороты по финансовым счетам, что ограничивает возможности планирования.

Разработанный модуль финансового планирования интегрирован в систему Navision, имеет тесную взаимосвязь с блоками финансового учета, управления закупками и продажами, производственного планирования.

Общая структура разработанной системы приведена на рис. 1. Прямоугольниками на схеме показаны справочники системы, с которыми работают пользователи, кружками — внутренние справочники, в которые собираются промежуточные данные, используемые для формирования бюджетов.

В блоке управления производством функции планирования реализованы посредством составления планов выпуска готовой продукции и полуфабрикатов, плана закупок сырья и плана загрузки производственных мощностей. План производства продукции формируется на основе прогнозов продаж, включающих как уже заключенные контракты на поставку продукции, так и перспективные заказы. На первом этапе планирования формируется план MPS, который затем может быть детализирован в соответствии с концепцией MRP II [6]. Помимо плана производства исходными данными для составления плана MRP служат спецификации продукции, описывающие

состав каждой единицы номенклатуры, и производственные маршруты, которые детализируют требования к производственным мощностям, необходимым для выпуска единицы продукции.

Учет в производственном модуле подразумевает:

- отслеживание состояния каждого производственного заказа (заказ может быть запланирован, утвержден, запущен на выполнение, завершен, отменен);
- учет операций расхода сырья и выхода готовой продукции (в том числе формирование соответствующих бухгалтерских проводок);
- поддержку актуальности данных по наличию доступных производственных мощностей.

В блоке финансового планирования перечисленные выше функции планирования, учета, анализа и регулирования реализуются посредством составления бюджетов, сбора фактических данных, план/фактного анализа и корректирования бюджетов на основании фактических данных. Схематично структура модуля показана на рис. 2.

Формирование бюджетов. В первую очередь перед формированием бюджетов строится модель финансовой структуры компании — *центры финансовой ответственности* (структурные подразделения предприятия, несущие ответственность за финансовые показатели в рамках своей области деятельности), различные их типы и иерархия подчиненности. На основании модели финансовой структуры разрабатываются бюджеты подразделений различных уровней и консолидированные бюджеты предприятия.

Сбор фактических данных. Сбор фактических данных в статьи бюджетов осуществляется со счетов и аналитических разрезов "Главной книги". Все хозяйственные операции в системе регистрируются в "Главной книге", следовательно, существует возможность автоматического текущего контроля исполнения планов.

Отчетность. Система содержит встроенные средства формирования отчетности. Структура отчетности может определяться как структурой бюджета, так и определенными пользователем аналитическими разрезами. Данные из системы могут быть выгружены в MS Excel. Дополнительно имеется возможность получения информации в графическом виде — система позволяет строить графики план/фактного анализа в MS Excel.

Контроль исполнения бюджета. Система позволяет контролировать исполнение бюджета путем сравнения плановых и фактических показателей. Отклонения рассчитываются с учетом аналитики, определенной при настройке бюджетной модели. В системе можно проводить анализ как по компании в целом, так и по ее различным филиалам и подразделениям. Дополнительно можно проводить сравнение различных версий бюджетов.

Объединение двух программных модулей в единую интегрированную систему позволило значитель-

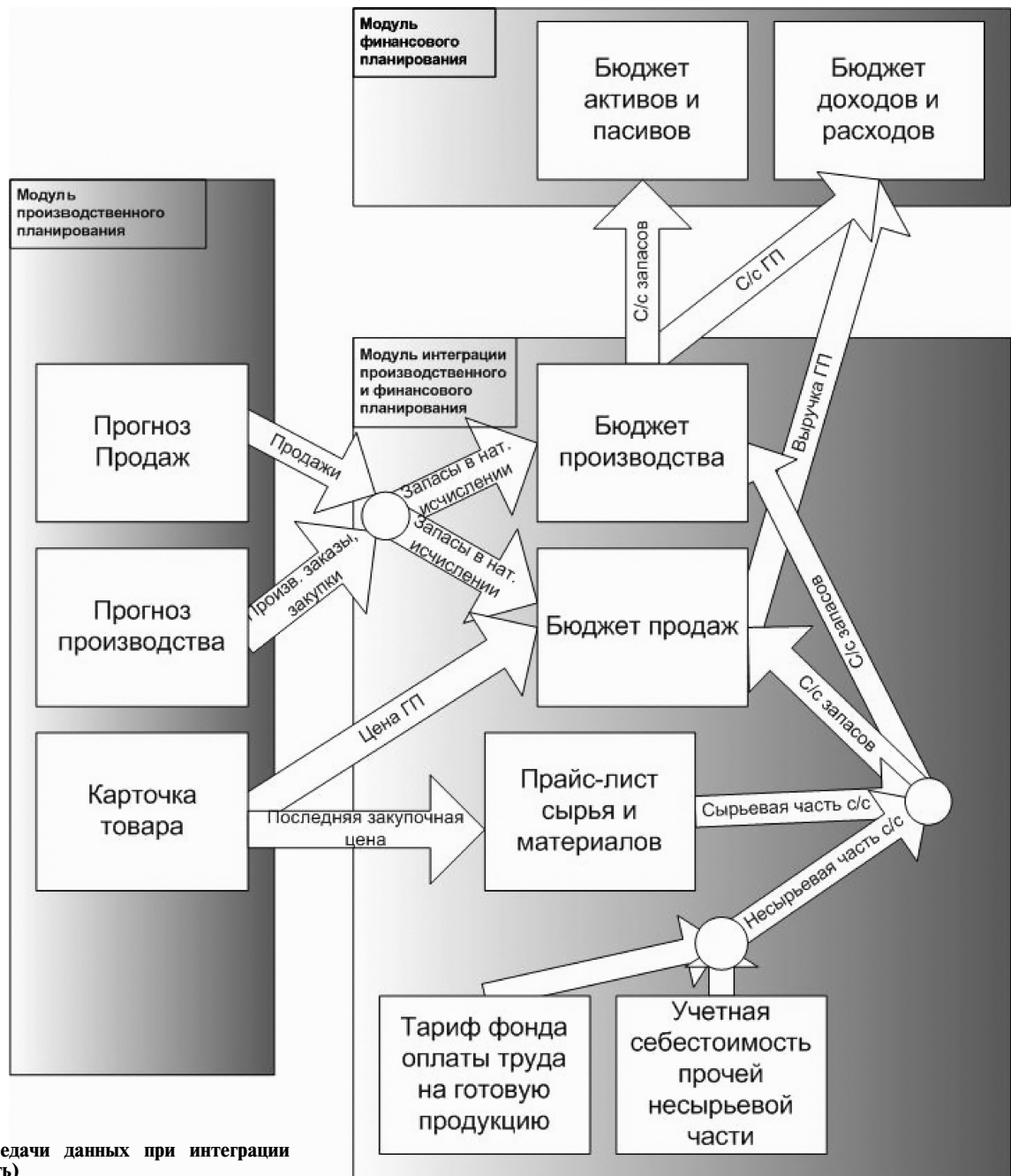


Рис. 1. Схема передачи данных при интеграции (с/с – себестоимость)

но расширить их функциональные возможности за счет более широкой интерпретации массивов данных (например, план загрузки производственных мощностей может рассматриваться в качестве бюджета производственного отдела, в свою очередь являющегося частью бюджета доходов и расходов предприятия; бюджет продаж может служить отправной точкой для формирования плана MPS и т. д.).

Результаты разработки и апробация

Реализация описанных в статье изменений позволила расширить функциональность системы, дополнив ее следующими возможностями:

- формирование бюджетов закупок сырья и материалов (вручную и на основе планов MPS и MRP);
- формирование бюджета продаж готовой продукции (вручную и на основе планов MPS и MRP);
- сбор фактических данных об объемах закупок, производства и продаж непосредственно из блоков управления закупками/продажами и производством без вмешательства пользователя;
- анализ исполнения производственных планов в натуральном и денежном исчислении;
- возможность оперативного анализа изменения финансовых показателей при внесении изменений в производственные планы;

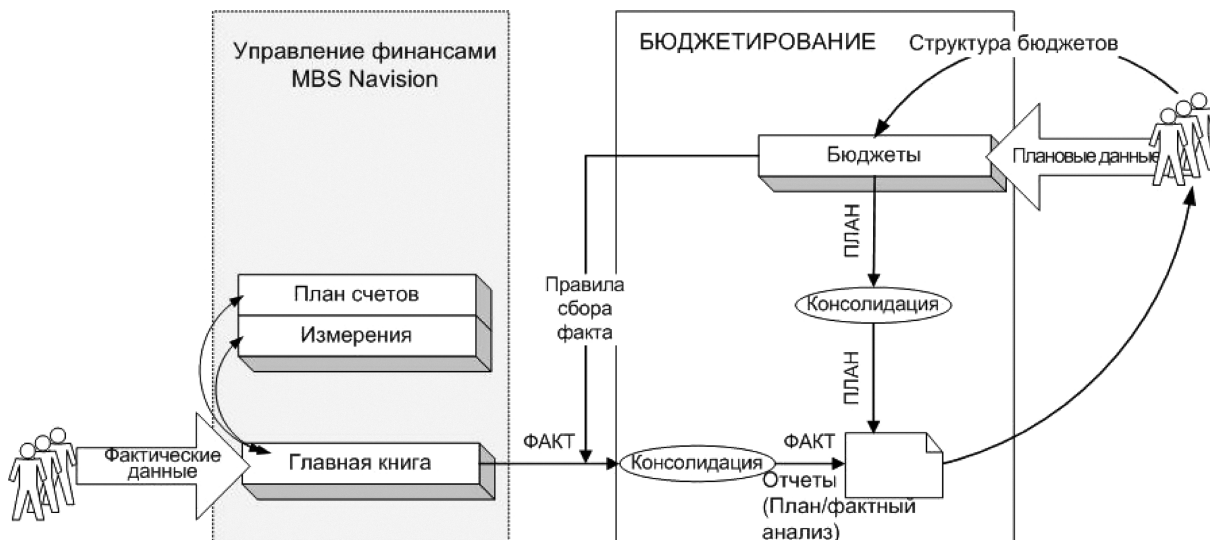


Рис. 2. Структура блока финансового планирования

- выгрузка/загрузка производственных бюджетов в файлы MS Excel для обмена данными с сотрудниками компании и контрагентами.

Формирование плана закупок сырья. Пользователи имеют возможность сформировать план закупок сырья на основе существующего плана производства. План закупок формируется в двух вариантах: *just-in-time*, который рассчитывается в соответствии с концепцией производства точно в срок исходя из предположения, что сырье должно быть (и может быть) доставлено точно к тому моменту, когда оно потребуется в производстве [7]; а также *логистический план*, учитывающий размер минимальной партии товара, *минимальное тарное место* (минимальное количество товара, которое может быть размещено на складе) и наличие складского запаса сырья, достаточного для обеспечения бесперебойного производства в случае возникновения проблем с доставкой.

Формирование плана продаж готовой продукции. На основании составленного прогноза производства создается план продаж готовой продукции, в который включается график исходящих платежей с учетом возможных отсрочек платежа за различные категории продукции. В отличие от заказов покупки, в которых сроки отгрузки и оплаты зависят от договоренности с конкретным поставщиком, соответствующие сроки заказов продажи зависят только от категории продукции.

Объединение функциональных модулей производственного планирования и бюджетирования позволило сократить время формирования плановых показателей бюджетов и сбора фактических данных. Механизм настройки бюджетов, связывающий бюджетные показатели с данными производственного модуля (такими как объемы вы-

пуска готовой продукции и полуфабрикатов, объемы закупок сырья, количество продукции в НЗП и т. д.), позволяет более оперативно формировать бюджеты, исключая необходимость сбора разрозненных данных производственного планирования. Кроме того, достигается возможность оперативного изменения источников данных бюджетов в случае изменения требований к ним.

В настоящее время блок финансового планирования в составе АСУ внедрен в промышленную эксплуатацию, выполняется сбор статистических данных о его применении для определения эффективности системы. Кроме того, подсистема финансового планирования АСУП Navision значительно расширена. В ее рамках реализована методология роллинг-бюджетирования, позволяющая корректировать бюджеты на основании полученных фактических данных; выполняются начальные этапы внедрения в эксплуатацию доработанной подсистемы.

Список литературы

1. Гительман Л. Д. Организация бизнес-планирования на предприятии [Электронный ресурс] // Электронная библиотека Института экономических преобразований и управления рынком: < http://www.socioego.ru/practica/organiz/planir/ch_1.html > (01.02.07).
2. Кляшторная О. Бюджетирование: старое понятие в новых формах // Директор ИС. 2006. № 3.
3. www.intalev.ru [Электронный ресурс] // web-сайт компании Инталев.
4. Глинских А. Особенности выбора ERP-системы [Электронный ресурс] // Информационный бюллетень Jet Info Online: < <http://www.jetinfo.ru/2002/2/1/article1.2.2002245.html> > (01.06.07).
5. Колесов А. Оценка масштабируемости и производительности "1С:Предприятия 8.0" // PC Week. 2004. № 9.
6. Гаврилов Д. Управление производством на базе стандарта MRP II. СПб.: Питер, 2002. 352 с.
7. Сергеев В. И. Менеджмент в бизнес-логистике. М.: Информационный издательский дом "Филинь", 1997. 772 с.

С. В. Синицын, канд. техн. наук, доц.,
О. И. Хлытчиев,

Московский инженерно-физический институт
(государственный университет)

От процессов предприятия к основным объектам системы документооборота программных проектов

Рассмотрен предложенный авторами подход к построению объектного описания системы документооборота программных проектов на основе процессного описания деятельности предприятия в виде IDEF0, а также построение и анализ модели документооборота такого предприятия на языке π -исчисления.

Введение

Практически любая компания, занимающаяся разработкой и верификацией программного обеспечения (ПО), рано или поздно сталкивается с необходимостью хранить и обрабатывать огромное количество информации. Чем сложнее проект, чем выше его критичность и значимость, тем выше требования к процессу управления данными. Более того, для обеспечения гарантии качества выходного продукта необходимо установить и поддерживать определенные процессы, регламентируемые деятельностью компании и отраслевыми стандартами. Особенно это важно для таких областей, как встроенное ПО, где цена ошибки может быть слишком велика или даже сопряжена с угрозой жизни людей (например, авиационное бортовое ПО).

Для структурирования, управления и анализа электронной информации используют различные виды систем документооборота, которые оперируют термином документ¹. Таким образом, для любой компании, связанной с разработкой и тестированием ПО, возникает необходимость внедрения системы документооборота. Такую систему можно специально разработать для конкрет-

¹ В данной работе под документом понимается единица информации, которая имеет набор свойств, набор ролей, которые могут работать с документом, и жизненный цикл, описывающий все стадии и процессы работы с этим документом. Документы, обладающие одинаковым жизненным циклом, одинаковыми ролями и свойствами будем называть документами одного типа, а совокупность характеристик документа — типом документа.

ной компании, или настроить, взяв за основу уже имеющуюся систему. В любом случае система поддержки документооборота — это система для предприятия, а не наоборот, поэтому все требования к ее функционированию заложены в процессе деятельности предприятия. А значит, первичная задача — это выявление и описание всех внутренних объектов системы в соответствии с основными процессами компании.

Формализация описания процессов и объектов предприятия

Формализованное процессное описание предприятия может уже существовать на момент разработки или внедрения системы электронного документооборота, или, если такого описания еще не существует, оно может быть легко построено, так как для будущих пользователей логично и понятно мыслить в терминах действий — определять, что они делают. Именно процессное описание задает первичные требования к системе документооборота, которые определяют типы документов, свойства и правила работы с ними. В то же время существует ряд стандартов, регламентирующих деятельность предприятий. Проверка соответствия предприятия этим стандартам упрощается при наличии формализованного описания его процессов, а некоторые стандарты, например ISO9001:2000 [1], прямо ориентированы на процессное описание.

Популярной формой представления процессной модели является язык IDEF0 [2, 3]. IDEF0-модель определяет набор процессов, работающих по определенным правилам и обменивающихся между собой типизированными данными, а всю структуру процессов можно представить в виде графа, у которого вершины представляют собой действия, а дуги — поток данных. Именно данные следует рассматривать как документы электронной системы, а вершины графа отражают возможные действия с документами.

Вместе с тем, основные типы документов можно рассматривать как классы, а сами документы — как объекты системы документооборота. В этом случае документы будут являться вершинами нового графа, а действия могут быть представлены как связи между различными документами или как преобразование отдельного документа. Более того, разработка архитектуры системы в настоящее время опирается на объектный подход, а саму архитектуру удобно описывать с помощью нотации UML [4]. Диаграмма классов достаточно точно описывает архитектуру системы документооборота и определяет основные типы документов.

Проблема построения архитектуры системы документооборота

Итак, в ходе рассмотрения описания деятельности предприятия должна быть построена архитектура будущей системы документооборота, но исходные данные — это функциональное описание, а необходимый результат — это объектное представление документов. Задача данного исследования состоит в том, чтобы построить алгоритм перевода одного описания в другое.

Конечно можно строить архитектуру с нуля, используя функциональное описание лишь в качестве неформальных требований и получая основную информацию от сотрудников (будущих пользователей). IDEF0-описание формируется лишь для тех подсистем, где это наиболее удобно [5]. В этом случае придется следить за тем, чтобы схема документооборота не только не противоречила процессной структуре деятельности организации, но и как можно полнее отражала ее. Дополнительной сложностью в данном подходе является необходимость общения с будущими пользователями в терминах объектной архитектуры, которые не всегда им понятны.

В работе предлагается использовать функциональное описание как формальные требования и строить архитектуру системы переводом одного описания в другое. Такое преобразование позволит построить объектную модель документооборота и одновременно сохранить весь опыт работы компании и не упустить никакие особенности, если преобразование будет полным. Общение с будущими пользователями в данном случае будет происходить только на понятном им "языке", но достаточно четко отражаться в архитектуре системы. Кроме того, в ходе построения объектного описания могут быть выявлены существующие противоречия в процессном описании, не замеченные ранее. Правильная организация процесса перехода от процессного описания к объектному может существенно ускорить и удешевить разработку или настройку системы документооборота.

Удобство использования языка IDEF0 для описания бизнес-процессов предприятия не вызывает сомнений [6, 7]. Однако задача перехода от процессов предприятия к основным объектам системы документооборота и соответственно к документам и их жизненным циклам является новой, и простого однозначного ее решения на данный момент не существует.

В работе предлагается решать такую задачу путем выделения основных объектов среди всех предметов IDEF0-модели и дальнейшего анализа этих объектов и их встречаемости в процессах для построения их жизненных циклов. Такой подход базируется на анализе IDEF0-модели и для его ус-

пешного применения, возможно, потребуется наложить определенные ограничения на саму исходную модель. В данной работе описывается только выделение основных типов объектов будущей системы без построения ее полной архитектуры, а определение жизненных циклов документов рассмотрено в работе [8].

Проблема терминологии и словарь проекта

Прежде чем перейти к алгоритму определения типов объектов, следует отметить, что в процессе построения IDEF0-модели системы обычно участвуют несколько разработчиков, поэтому неизбежно возникает расхождение в употреблении одних и тех же терминов. Проблема терминологии может потерять или исказить смысл диаграмм, и для ее решения предлагается использовать средства поддержки общего терминологического словаря проекта. Общий словарь не только обеспечивает возможность установления связи между терминами, но и облегчает последующее выделение объектов в операционном описании.

Словарь терминов изменяется как синхронно с диаграммами (внесение изменения в диаграмму, построение новых диаграмм модели), так и асинхронно (создание базового словаря, импорт словаря другого проекта). При старте любого проекта в начальном состоянии определены всего несколько терминов, которые планируется использовать в модели. Возникает проблема формирования некоего начального множества терминов для их последующего анализа и использования.

Одним из подходов к созданию первичного терминологического словаря может служить построение небольшого набора основных диаграмм без использования словаря, а затем, после построения общей "схемы" системы (первых основных моделей), формирование словаря из использованных в диаграммах терминов. В таком случае в начальной версии словаря не будет неиспользуемых терминов и отпадет необходимость в дополнительном времени для выделения основных терминов на начальном этапе. Кроме самого термина, в словарь удобно заносить его тип — активность или дуга, а также сведения о вхождении одного предмета в другой — для разъединяющихся или соединяющихся дуг.

Алгоритм преобразования описаний

Основная идея преобразования построенной IDEF0-модели в диаграмму классов UML-модели заключается в выделении объектов в операционном описании. Объекты не видны явно, однако основные сведения о них заложены в дугах модели. Для достаточно глубоко детализированной модели (глубина детализации средней модели

~ 6—7) число дуг примерно равно 700...1500. Анализировать такое число дуг возможно только с помощью некоторых формальных алгоритмов, основной задачей которых будет объединение дуг в объекты с учетом их прохождения через активности IDEF0-модели. Число получаемых на выходе объектов для приведенной системы будет колебаться от 10 до 15.

Общая последовательность действий может быть выражена следующим образом.

- Первый шаг — построение таблицы предметов-операций. Для этого выделяются все дуги с одинаковым именем. Для каждого такого множества определяются все операции, в которых участвует хотя бы одна дуга из рассматриваемого множества дуг с одинаковыми именами. Эти операции заносятся в таблицу, для каждой операции указываются все ее связи с дугами и типы этих связей.
- Второй шаг — расчет частот встречаемости для каждого предмета по таблице предметов-операций. Частоты встречаемости определяются путем подсчета числа операций, связанных с рассматриваемым множеством дуг.
- Третий шаг — сортировка предметов по частоте встречаемости.
- Далее начинается последовательное рассмотрение каждого предмета IDEF0-модели, начиная с того предмета, частота встречаемости которого наименьшая.
- Проверяется возможность объединить предмет по ключевому слову. Алгоритм ключевого слова заключается в объединении всех предметов, имеющих одинаковое ключевое слово, которое определяется на стадии занесения названия дуги в терминологический словарь проекта. Ключевое слово обычно совпадает с существительным — названием сущности ("утвержденная заявка" — "заявка"), но возможны и другие варианты ("подтверждение заявки" — "заявка"). Таким образом, если ключевой термин в словаре для значений дуг одинаков, то эти дуги объединяются.
- Если объединение не удалось провести по первому методу, то переходим к двум следующим методам. Они всегда проверяются вместе и выдают некие веса, которые затем анализируются совместно.
- Проверяется возможность объединить предмет с помощью функционального метода. Этот метод заключается в определении для каждой операции основного предмета, над которым она проводится по таблице предметов-операций. Для каждой операции в таблице рассматриваются все предметы, являющиеся входами и выходами одновременно, и среди них выбирается "основная" (операции, для которых не

существует предмета, являющегося одновременно входом и выходом, можно всегда определить как операции типа "удаление", "выделение свойства" или операции "соединения" основного объекта). Считается, что операция может быть проведена только лишь над одним предметом. Все случаи, когда операция выполняется над несколькими предметами, сводятся к случаю создания/удаления одного предмета с участием других предметов. Далее оценивается возможность объединения редко встречающейся дуги в диаграммах с той дугой, в операциях с которой наиболее часто встречается рассматриваемая дуга. В зависимости от количества операций над предметом, которые можно свести к операциям над предметом, с которым предполагается объединение, рассчитывается вес данного метода объединения.

- Далее проверяется возможность объединить предмет с помощью семантического метода, который заключается в оценке возможности объединения разветвленных и соединенных дуг. Непомеченные ветви содержат все предметы, указанные в метке дуги перед разветвлением, представляют по сути один объект, и могут быть объединены. "Разделяющие" предметы дуги часто тоже можно объединить, но только, если ответвившаяся дуга не рассматривается как отдельный, часто используемый предмет. Их относительные частоты встречаемости можно рассматривать как вес этого метода.
- Складываются веса, полученные на предыдущих двух шагах, если сумма превышает коэффициент объединения для этих методов (данный параметр задается априорно, при определении алгоритмов расчета весов), то предметы объединяются.
- Далее переходим к началу алгоритма, к построению новой таблицы предметов-операций (или подсчету частот), которая изменяется, если было проведено объединение.

На каждом шаге можно вычислять функцию "частотного качества", которая может быть, например, следующего вида:

$$F = (\Delta a_1)^2 + (\Delta a_2)^2 + \dots + (\Delta a_n)^2,$$

где Δa_k — разность числа встречаемости в таблице предметов-операций двух объектов, вычисляемая для всех возможных пар. Чем меньше значение этой функции, тем более равномерно моделируемая система разбита на объекты.

Пи-исчисление — язык для описания параллельных процессов

Предложенный в работе метод преобразования операционного описания в объектное не является

в достаточной степени формальным, кроме того, он делает затруднительным создание средства, помогающего провести такое преобразование, так как требует анализа графических диаграмм и терминологии. Попробуем более строго и удобно для автоматического анализа определить процессное описание с помощью пи-исчисления.

Язык пи-исчисления [9, 10, 11] является достаточно молодым, был специально создан для описания параллельных процессов и уже начал успешно использоваться при описании бизнес-процессов [12, 13].

Язык пи-исчисления оперирует терминами "переменная", "канал" и "процесс". Взаимодействие процессов описывается через передачу данных (переменных) по каналам (имеющим имена), причем в качестве данных можно передавать каналы (это позволяет передавать "ссылки" на процессы). Передача данных y по каналу x описывается как $\bar{x}y$, а прием значения y из канала x записывается как $x(y)$. Во второй записи переменная y является связанной и при приеме фактического значения заменяется на это значение.

Например, процесс A , передающий значение x по каналу c , и процесс B , принимающий данные из канала c в переменную y могут быть представлены следующим образом:

$$A = \bar{c}x.A_1; \quad B = c(y).B_1.$$

Вместе они могут быть записаны следующим образом:

$$(\nu c)(\bar{c}x.A_1|c(y).B_1).$$

Для этого выражения может быть применена операция редукции для моделирования взаимодействия этих процессов. Для приведенного выше примера на первом шаге редукции по каналу c будет передано значение x , соответственно все значения y будут заменены на x . Процесс B прочитает из канала c значение x и далее редукция станет более невозможной. Формальные определения пи-исчисления могут быть найдены в работах [9, 10].

Переход от IDEF0-моделей к пи-исчислению

Итак, в качестве исходных данных рассматривались IDEF0-диаграммы, работающие непосредственно с процессами предприятия, для которых всегда справедливо то, что они выполняются или могут выполняться параллельно. На основе этих описаний и будем строить модели на пи-исчислении.

Рассмотрим взаимодействие двух процессов в виде IDEF0-описания (рис. 1).

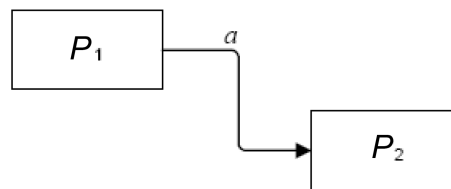


Рис. 1. Простое взаимодействие двух процессов

На языке пи-исчисления это может выглядеть следующим образом:

$$P_1|P_2,$$

или

$$\bar{a}y.P_1|a(x).P_2.$$

По каналу a процесс P_1 передает данные, а процесс P_2 получает их. Так как для IDEF0 не существует единичной передачи данных, а они могут передаваться, когда это необходимо, следует рекурсивно определить P_1 и P_2 следующим образом:

$$P_1 = ay.P_1 + \tau.P_1;$$

$$P_2 = a(x).P_1 + \tau.P_2.$$

Тогда общий процесс будет иметь вид:

$$(\nu a)(\bar{a}y.P_1 + \tau.P_1|a(x).P_1 + \tau.P_2).$$

Здесь a — связанная переменная, а в контексте IDEF0-диаграммы a — это предмет или общее обозначение данных, передаваемых от P_1 к P_2 .

В общем случае процесс IDEF0, показанный на рис. 2, можно представить следующим образом:

$$P = Pin|Pout|Pmech|Pctrl,$$

где Pin — процесс потребления входов; $Pout$ — процесс посылки выходных данных; $Pmech$ — процесс получения механизмов; $Pctrl$ — процесс получения контрольных воздействий.

Сведения о внутренней структуре этих четырех процессов можно получить лишь на следующем уровне детализации при рассмотрении внутренней структуры процесса P .

Для всех процессов, потребляющих данные, можно использовать следующее рекурсивное представление (на примере Pin):

$$Pin = i(x).Pin + \tau.Pin,$$

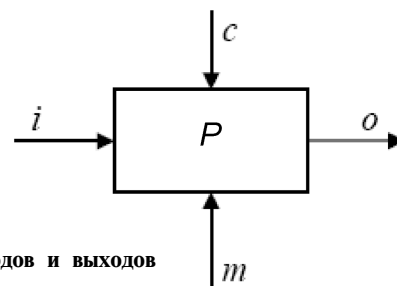


Рис. 2. Общий вид входов и выходов процесса

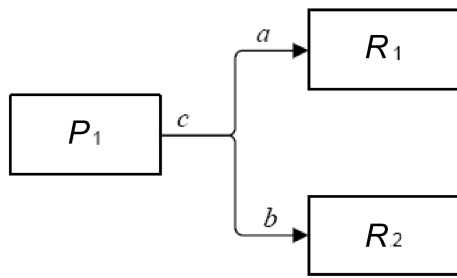


Рис. 3. Разветвление дуги

а для процесса Pout

$$P_{out} = \bar{i}y.P_{out} + \tau.P_{out}.$$

Это представление получено при непосредственном использовании ICOM-механизма IDEF0. Для процесса связанными переменными в данном случае будут i, c, m, o .

В диаграммах IDEF0 дуги могут разделяться и сходиться. Это может быть обусловлено как передачей одних и тех же данных в несколько процессов, так и "разделением" ("сборкой") данных. В первом случае на языке пи-исчисления это представляется недетерминированной передачей одних и тех же данных по разным каналам. Во втором случае следует рассмотреть разделение и соединение данных.

Рис. 3 показывает разветвление дуги c на две дуги — a и b .

Рассмотрим функцию разделения данных в виде процесса Div:

$$P_{1out} = (vc)(\tau.P_{1out} + \bar{c}x.P_{1out}|Div(c));$$

$$Div(c) = (c(y).\bar{a}y.0 + c(z).\bar{b}z.0).$$

При рассмотрении процессов P_1, R_1 и R_2 вместе переменные c, a и b будут связанные, функция Div более аккуратно должна определяться как функция трех переменных: $Div(c, a, b)$.

Аналогично рассматриваются слияния дуг и построение функции-процесса Add. Туннельные дуги диаграмм IDEF0 здесь не рассматриваются, но они тоже могут быть однозначно определены на пи-исчислении.

На основе описанных выше построений не сложно показать, что любую IDEF0-модель, любой степени детализации можно описать на языке пи-исчисления. Если четко следовать предложенному методу, то это преобразование будет однозначным и полным с точностью до имен несвязанных переменных.

Построенная модель процессов предприятия позволяет провести моделирование этого пред-

приятия путем редукции выражений пи-исчисления.

Алгоритм преобразования для пи-исчисления

На основе языка пи-исчисления можно строить описания для автоматического анализа диаграмм, и более того, для задания описания процессов предприятия в электронной системе, использующей логику этих процессов для информационной поддержки. В качестве примера такой системы можно отметить систему документооборота, которая как раз должна поддерживать электронные документы, функционирование которых описывается основными процессами предприятия. Пример языка, построенного на основе пи-исчисления, можно найти в работах [12, 14].

Анализ полученного описания процессов предприятия на языке пи-исчисления заключается в рассмотрении связанных переменных. Для начала они объединяются в группы на основе словаря, согласно первому шагу описанного выше алгоритма преобразования IDEF0-модели в основные объекты будущей системы. Это объединение может выполняться уже в ходе построения пи-модели.

Дальнейшее объединение рассматривает функции-процессы Div и Add. Для их входов и выходов рассчитываются частоты встречаемости, как по отдельности, так и при замене одного из входов одним из выходов (объединение связанных переменных входа и выхода). При получении относительно низких частот встречаемости для необъединенного случая выбирается то объединение переменных входа и выхода, которое приводит к более сильной разнице результирующих вариантов частот встречаемости, т. е. более "сильная" переменная "притягивает" более "слабую". Для оценки получаемого результата можно использовать критерий "80 % к 20 %".

Рассмотрение процессов позволит определить основной объект процесса — в большинстве случаев это переменная, присутствующая как во входах (i) процесса, так и среди его выходов (o). Связанные переменные с маленькой частотой встречаемости, но относительно часто используемые с одним и тем же основным объектом процесса как вход или выход, следует объединять.

Представление процессной модели предприятия на языке пи-исчисления позволяет задать жизненный цикл документа на первичном этапе, путем передачи порождающим документ процессам ссылок на все процессы, затрагивающие этот документ. Вместе с тем, можно "растворить" жизненный цикл документа среди процессов, т. е. заранее не указывать все возможные процессы "пользователи", но внутри каждого процесса оп-

ределить все возможные каналы передачи документа. Анализ этих подходов выходит за рамки данной статьи.

Заключение

Таким образом, при необходимости построения или настройки системы документооборота можно пойти путем выделения основных типов документов из процессного описания предприятия. Предлагаемые в работе методы были успешно применены при проектировании системы документооборота для одной из компаний, специализирующейся на разработке и верификации встроенного ПО. На основе примерно 40 процессов и порядка тысячи предметов были выделены основные типы документов и построены их жизненные циклы. Эксплуатация электронной системы документооборота, в основе которой были положены выделенные типы документов, показала применимость описываемых методов и алгоритмов. Кроме того, после внедрения системы документооборота это предприятие успешно прошло сертификацию по таким отраслевым стандартам как ISO9000:2000 и AS9100 [15].

Список литературы

1. ISO 9001:2000 Quality Management System — Requirements. ISO. 2000. 33 p.
2. Марка Д. А., МакГоуэн К. Методология структурного системного анализа и проектирования SADT. М.: Метатехнология, 1993. 240 с.

3. Р 50.1.028—2001. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. Постановление Госстандарта России № 256-ст Р от 02.07.2001 № 50.1.028-2001.

4. Фаулер М., Скотт К. UML основы. Краткое руководство по унифицированному языку моделирования. М.: Символ-Плюс, 2001. 192 с.

5. Turk Z. Object-Oriented Modelling and Integrated CAD: Automation in Construction. Vol. 1. 1993. P. 323—337.

6. Ang C. L., Khoo L. P., and Gay Robert K. L. IDEF* — A comprehensive Modelling Methodology for the Development of Manufacturing Enterprise Systems // International Journal of Production Research. 20 November 1999. Vol. 37. Number 17. P. 3839—3858 (20).

7. Mustafa M. R., Hossam I. New product development: Assessment tools // Journal of Modelling in Management. 2007. Vol. 2.

8. Синицын С. В., Хлытчиев О. И. Переход от процессного описания функционирования предприятия к жизненным циклам документов // Труды XV научно-технического семинара. Сентябрь 2006 г., Алушта. М.: МИФИ, 2006. С. 100.

9. Milner R., Parrow J., Walker D. A Calculus of Mobile Processes — Part I. LFCS Report 89—85. University of Edinburgh. June 1989. 46 p.

10. Milner R., Parrow J., Walker D. A Calculus of Mobile Processes — Part II. LFCS Report 89—86. University of Edinburgh. June 1989. 41 p.

11. Фиошин М. Основы пи-исчисления (<http://progr.tsi.lv/research/picalc.pdf>).

12. Benjamin C. Pierce Programming in the pi-calculus: A tutorial introduction to Pict. 1997. (<http://citeseer.nj.nec.com/pierce97programming.html>).

13. Havey M. BPM Theory for Laymen // Webservices journal. 2005. Vol. 5. Iss. 5. (<http://webservices.sys-con.com/read/issue/556.htm>).

14. Howard S., Workflow P. Is just a Pi process. January 2004. Fingar. 36 p. (<http://www.bptrends.com>).

15. AS 9100A. Quality System — Model of Quality Assurance in Design, Development, Production, Installation and Servicing. G-14 Americas Aerospace Quality Group (AAQG). SAE. 2003. 69 p.

ДИСКУССИОННЫЙ КЛУБ

УДК 004.31

А. Я. Фридланд,

Тулский государственный педагогический университет им. Л. Н. Толстого (ТГПУ)

О сущности информации: два подхода

Рассматриваются две гипотезы о сущности информации — атрибутивный и антропологический подходы, показаны следствия их принятия в обучении информатике. Предложен подход к информации как сущности, состоящей из двух компонентов: смысла, генерируемого человеком и не отторгаемого от него, и данных, отторгаемых от человека в разных формах.

Этой статьей мы вступаем в дискуссию о сущности информации, открытой в журнале [1]. В статье рассматриваются две гипотезы о сущности информации, показаны следствия их принятия в обучении информатике. Не будем останавливаться на вопросе, кому и зачем нужно понимать сущность информации, использовать термин и понятие. Отметим лишь существенное влияние понимания этого явления на сознательный выбор каждого человека между научным мировоззрением и мракобесием, широко представленным в современном обществе. Предоставим читателям самим сделать вывод, какая гипотеза более приемлема.

Предварительно несколько слов о методологии.

1. Информация, как любой объект или явление, имеет три составляющие: сущность, опреде-

ление и термин. У любого явления есть нечто объективное — это сущность. У сущности может быть много определений, более или менее точных, в зависимости от состояния изученности или цели человека, дающего определение. Представителем сущности и, следовательно, определений есть термин.

2. Следствия употребления термина не должны быть противоречивы.

3. Определения одного явления не должны быть принципиально разными для отдельных наук и бытового употребления (совместимость по вертикали),

4. Разные уровни определения должны быть между собой согласованы, определения для школьников и специалистов должны отличаться уровнем формальности, а не разным смыслом (совместимость по горизонтали).

5. "Имеются два типа определений: 1) логически строгое сведение определяемого объекта к уже введенным понятиям; 2) описательное определение с помощью слов разговорного языка" [2, с. 8]. Большинство определений в информатике относятся к описательным определениям.

6. Необходимо следовать принципу бритвы Оккама: "Сущности не следует умножать без необходимости" [3, с. 317].

Первая гипотеза — атрибутивная

В изложении гипотезы будем опираться на работу К. К. Колина [1]: "Сущность *атрибутивного* подхода заключается в том, что информация предполагается неотъемлемым свойством (атрибутом) материи и поэтому она может проявлять себя во всех объектах, процессах и явлениях как живой, так и неживой природы" [1, с. 64].

Такое серьезное утверждение требует доказательств, вместо этого декларируется, что атрибутивный подход "*в последние годы завоевывает*¹ в научной среде все большее количество сторонников" [1, с. 64], что "*хорошо известно, что информация является основой функционирования любых организованных систем, а не только объектов живой природы. Не зря же даже совокупность вращающихся вокруг Солнца планет мы называем Солнечной системой. Именно системой, в которой между ее отдельными элементами существуют как прямые, так и обратные информационные связи*" [1, с. 65].

Можно ли такие заявления считать доказательством? Здесь можно увидеть много противоречий, остановимся на трех.

¹ Здесь и далее выделение в цитатах полужирным шрифтом, где не оговорено противное, — мое, А. Ф.

1. Определение системы совсем не требует наличия информационных связей.

2. Наличие информации обосновывается наличием информационных связей, хотя еще не было определения ни того, ни другого.

3. Нарушен принцип бритвы Оккама, Солнечная система полностью описывается гравитационными взаимодействиями. Именно об этом писал академик Н. Н. Моисеев: "*Если же речь идет о механизмах или структурах взаимодействия, которым нельзя приписать целенаправленные действия, ... то здесь можно обойтись без понятия "информация". Конечно, при желании движение точки в гравитационном поле можно трактовать и следующим образом. Пусть точка А переместилась из положения x в положение x' . Она получила "информацию" о том, что напряженность гравитационного поля изменилась: вместо $u(x)$ она стала равной $u(x')$. В результате выработалось новое значение действующей силы, изменяющее величину ускорения. Однако такое "расширение" по существу ничего не дает. Имеется вполне четкое взаимодействие между параметрами движения материальной точки и полем, потенциал которого определяется величиной $u(x)$. И это взаимодействие полностью описывается законом всемирного тяготения, а характер движения является прямым следствием закона сохранения импульса. Информационные процессы здесь не при чем*" [4, с. 81—82].

В следующем абзаце К. К. Колин приводит еще одно соображение, которое должно свидетельствовать в пользу атрибутивного подхода: "*В монографии академика Б. Б. Кадомцева "Динамика и информация" убедительно показано, что весь окружающий нас мир представляет собой не что иное, как совокупность информационно открытых систем, которые непрерывно взаимодействуют между собой в процессе своей эволюции. При этом информация пронизывает все уровни организации материи — от квантовых систем до мегагалактических образований. Именно она определяет направление развития всех эволюционных процессов во Вселенной, является той главной первопричиной, под воздействием которой и осуществляется эволюция*" [1, с. 65].

Здесь не место детального рассмотрения работы Б. Б. Кадомцева, лишь отметим, что, во-первых, само слово "информация" Б. Б. Кадомцев в большинстве случаев употребляет в шенноновском смысле; во-вторых, вводится новое понятие "информационная связь": "*Главное, что характерно для предметов нашего окружения, это их постоянная информационная связь. Вся природа купается в лучах солнечного света, переливаясь красками, и этого уже достаточно, чтобы живые существа ве-*

ли постоянное наблюдение за своим окружением. Аналогичная связь через свет **может существовать и между объектами неживой природы**. С точки зрения квантовой механики это означает, что положение макротел постоянно **"измеряется"** окружающими их живыми и неживыми телами" [5, с. 78]. Обратите внимание на предположение, а не уверенность, Б. Б. Кадомцев пишет: "может существовать". Кроме понятия "информационная связь", Б. Б. Кадомцев использует понятия: "информационное поведение", "свобода воли": **"При взаимодействии систем с памятью и информационным поведением могут развиваться процессы адаптации одних систем к другим. Это аналог появления "знаний" о реакции других систем на поведение данной. Дальнейшее развитие механизмов адаптации и оптимизации поведения системы приводит к возможности обработки информации, т. е. к появлению элементов сознания. Но кроме свободы у сложных систем появляются зачатки воли"** [5, с. 345].

Продолжая рассуждения К. К. Колина и Б. Б. Кадомцева, можно прийти к выводу, что стул знает, что он стул, и даже знает, кто на нем сидит.

Следует отметить, что Б. Б. Кадомцев считает "возможным обработку информации" только "элементами сознания", что слово "знание", автор берет в кавычки, а слово "воля" использует в прямом значении. Чувствуя некоторую неубедительность своих аргументов, которые, как считает К. К. Колин "убедительно показывают ...", Б. Б. Кадомцев замечает: **"... книга содержит некоторые новые, подчас непривычные рассуждения. Кому-то они могут показаться неубедительными, а подчас и просто неправильными. Но я рекомендую читателям не делать поспешных суждений: некоторые вопросы обсуждаются повторно или даже несколько раз с нескольких иных точек зрения, и убедительность соответствующих выводов становится больше и больше"** [5, с. 17].

Нам не показались убедительными приведенные рассуждения, потому что все описанные в книге явления, по нашему мнению, можно описать в терминах взаимодействий систем, как и отмечал Н. Моисеев, без использования антропоморфных терминов.

Общепринято положение, высказанное академиком А. Мигдалом: **"В науке давно уже принято очень простое и убедительное предположение, что процессы, происходящие в живой природе, определяются, в конечном счете, законами взаимодействия электронов, атомов и молекул, установленными в физических экспериментах. До сих пор эта гипотеза подтверждалась. Более того, она оказалась настолько плодотворной, что привела к объяснению**

даже такого таинственного явления как наследственность" [6, с. 19].

Но где доказательства того, что из этого утверждения следует обратное: психические процессы, которые свойственны живым объектам, присущи неживым объектам?! В этом случае А. Мигдал предлагает использовать формулу научного подхода: **"хочется верить", но, раз "нет оснований", надо от этой веры отказаться"** [6, с. 14]. О компьютерах, мышлении и законах физики см. работу [7].

От сущности атрибутивного подхода к информации К. К. Колин переходит к физической сущности феномена информации, давая определение: **"Информация в широком понимании этого термина представляет собой объективное свойство реальности, которое проявляется в неоднородности (асимметрии) распределения материи и энергии в пространстве и времени, в неравномерности протекания всех процессов, происходящих в мире живой и неживой природы, а также в человеческом обществе и сознании"** [1, с. 65].

Из этой гипотезы получаются следствия, из которых мы рассмотрим лишь основные.

1. **"В однородных средах и в равномерно протекающих процессах информация отсутствует"** [1, с. 66].

Рассмотрим это свойство подробно на примере, приведенном самим К. К. Колиным: **"Возьмем обычный детский надувной резиновый шарик, надуем его и в таком виде привяжем на нитке к какому-нибудь предмету. При отсутствии внешних воздействий шар будет висеть неподвижно, так как давление молекул воздуха внутри него равномерно распределено по внутренней поверхности**

В данном случае мы имеем дело с однородным распределением вещества во внутреннем пространстве шара, т. е. с однородной средой, в которой отсутствует информация" [1, с. 68].

Во-первых, возникает вопрос: как правильно говорить, информация отсутствует **в шарике**; или информация **о шарике** отсутствует; или информация отсутствует в воздухе, заключенном в шарике, или шарик не имеет информацию? Можно считать этот вопрос несущественным, если удастся ответить на вопрос: стул обладает информацией? А книга, лежащая на столе? Книга, лежащая на столе, — это однородная среда, она неравномерно давит на стол, ее вес и есть информация?

Во-вторых, как это связано с определением "физической информации": **"... в последние годы быстро возрастают объемы научной информации о состоянии поверхности и недр нашей планеты, ближнего и дальнего космоса, частиц микромира, химических соединений и структурных элементов**

живой природы. Эта так называемая "физическая информация"... [1, с. 63].

То есть все, что раньше называлось физическими параметрами (характеристиками) теперь называется "физической информацией". Как быть с размерами шарика, — это информация?

В-третьих, как это согласуется с определением, приводимым К. К. Колиным: "... информационные ресурсы понимаются как документы или базы данных, содержащие некоторую информацию, а сама информация — как содержание этих документов или баз данных" [8, с. 97].

2. Информация подразделяется на физическую и идеальную: "Информация неразрывно связана с материей и энергией, которые являются ее носителями. Она представляет собой их атрибут, т. е. неотъемлемое свойство. Поэтому данный тип информации может быть назван "физической информацией" в отличие от "идеальной информации", которая является результатом деятельности сознания и о сущности которой следует поговорить особо" [1, с. 66].

К сожалению, автор не возвращается далее к "идеальной информации", однако из предыдущего абзаца следует, что это вообще не информация: "... информация — это не плод нашего воображения, не продукт деятельности сознания, а реальный физический феномен, характеризующий состояние и движение материи или энергии" [1, с. 66].

В другой статье все же допускается информация в "нашем сознании", но при определенных условиях: "... никакой информации в нашем сознании не возникнет, если ее нет в объективной реальности, которую это сознание наблюдает. Информация в сознании вторична — это всегда отражение некоторой первичной информации, уже существующей в объективной реальности" [9, с. 86].

Если понимать под объективной реальностью все, что создано без участия человека, то практически все объекты, которые создал человек, не наблюдаются в объективной реальности до их создания, например, колесо, автомобиль, самолет и т. п.

3. "Одним из важнейших свойств информации является ее относительность" утверждает К. К. Колин: "... для того, чтобы обнаружить информацию, необходимо иметь возможность сравнения наблюдаемого состояния некоторого объекта, процесса или явления с другим, уже известным из опыта предыдущих наблюдений, их состоянием.... Поскольку же различные наблюдатели одного и того же объекта, процесса или явления могут иметь свой собственный опыт предыдущих наблюдений, отличный от опыта других наблюдателей, то и их возможности по выявлению новой информации могут существенно образом различаться. Именно это и обу-

славливает относительность воспринимаемой ими информации. Таким образом, способности человека, технической системы или объекта к восприятию поступающей к ним из внешнего мира информации зависят не только от характера этой информации, но также и от того внутреннего тезауруса, который у них уже сформирован на основании накопленного ранее опыта своего функционирования и наблюдения за внешним миром. Этот вывод является исключительно важным результатом теории информации, который служит не только философской основой теории познания, но также и одним из основополагающих принципов создания интеллектуальных систем" [1, с. 66—67].

Из этой цитаты следует, что, во-первых, для "обнаружения информации" требуется человек ("наблюдатель"), и тогда не ясно как быть с объективностью; во-вторых, не только человек, но и техническая система (например, автомобиль), физический объект (например, булыжник на дороге) имеют тезаурус, "который у них уже сформирован на основании накопленного ранее опыта своего функционирования и наблюдения за внешним миром". Возникает противоречие с понятиями: "**Воспринять** — 1. Ощутить, распознать органами чувств. 2. Понять и усвоить" [10, с. 99]. "**Тезаурус** — 1. Словарь языка, ставящий задачу полного отражения всей его лексики. 2. Словарь или свод данных, полностью охватывающий термины, понятия какой-нибудь специальной сферы" [10, с. 791].

4. "Информация порождает движение материи и энергии в пространстве и времени" [1, с. 68].

Справедливость этого утверждения автор доказывает опытом, описанным в Свойстве 2. В надутым резиновом шарике делается прокол, и шарик летит в противоположную от прокола сторону: "... неоднородность, созданная в оболочке шара (отверстие), сразу же приводит к возникновению неоднородности в распределении молекул газа внутри него, и в результате этого начинается движение материи.... Образно выражаясь, можно утверждать, что информация порождает движение, а движение, в свою очередь, порождает время. Однако необходимо подчеркнуть, что первичным фактором в этой цепи причинно-следственных связей все же является именно информация" [1, с. 68].

Относительно этого свойства следует отметить, что опять нарушен принцип бритвы Оккама, реактивное движение совершенно адекватно описывается традиционными методами физики, привлечение новых терминов не приводит к дополнительному знанию.

5. Понятия "материя", "энергия", "информация" — равнозначны. Рассмотрим утверждение: "Понятия "материя", "энергия" и "информация" явля-

ются равнозначными по своему уровню общенаучными философскими категориями" [1, с. 69].

Это утверждение противоречит следующей цитате: "Сущность *атрибутивного подхода* заключается в том, что информация предполагается неотъемлемым свойством (атрибутом) материи" [1, с. 64].

Разве может "свойство" объекта быть равнозначным самому объекту? Противоречие возникает между утверждением о "равнозначности категорий" и утверждением, что информация есть свойство материи. Смотрим Толковый словарь: "*Свойство — качество, признак, составляющий отличительную особенность кого-чего-нибудь*" [10, с. 704].

6. Что является объектом изучения информатики? "*Объектами изучения информатики как фундаментальной науки являются основные свойства информации, закономерности процессов информационного взаимодействия в природе и обществе, а также методы организации этих процессов в технических, биологических и социальных системах*" [1, с. 69].

Это уже не свойство информации, а логичный вывод из предыдущих свойств и определения "информации", предложенного К. К. Колиным. Информатика в этом случае (раз уж она занимается даже реактивным движением) представляется в виде науки наук. В другой своей работе К. К. Колин пишет: "*Предмет информатики активно обсуждается в настоящее время так же, как и само понятие информации. В связи с этим в последние годы появились такие новые названия научных дисциплин, занимающихся изучением проблем информации, как информология и информациология*" [9, с. 88].

Специалистам хорошо известна информациология, но она больше похожа на религию, чем на науку. Приведем одну выдержку из известной книги: "*Информация — первооснова Мира. ... Информация — внутри нас и — вне нас. Все, что нас окружает, везде и всюду — информация. ... Информационное единство Вселенной подтверждается универсальными законами мироздания. Информация первична. Материя вторична*" [11, с. 483].

Председатель комиссии по лженауке при Президиуме РАН академик Э. П. Кругляков отнес "информациологию" к лженаукам [12].

Читателю предлагается самостоятельно сравнить приведенную выше цитату и утверждение К. К. Колина: "*Именно информация побуждает материю и энергию к движению, является его фундаментальной первопричиной*" [1, с. 67].

К. К. Колин определяет информологию так: "*... наука будущего, объектом изучения которой являются информационные ресурсы общества, а предметом — основные свойства этих ресурсов, закономерности их формирования и эффективность использования*" [8, с. 341].

Завершая анализ атрибутивного подхода к пониманию информации, мы хотим отметить, что если использовать этот подход, то остается неясными:

- возможности перехода к тем определениям информации, которые имеются в учебной и научной литературе;
- возможности связи с другими понятиями современной науки;
- связи с обыденным употреблением слова информация.

Гипотеза о субъективности информации, о ее принадлежности только высокоорганизованным живым объектам

Теперь перейдем к рассмотрению другой гипотезы, основанной на разделении понятия "информация" на два — "данные" и "смысл". И так как смысл в этой паре является главным, то именно смысл назовем "информацией".

Для определения сущности явления, которое понимается под термином "информация", обратимся к истории употребления термина "информация".

Историко-этимологический словарь утверждает, что слово "информация" происходит от латинского "*informātiō*" — "сообщение", "разъяснение", производное от "*informō*" — "придаю вид", "формирую", "организую", "обучаю", "воспитываю", "**мыслю**" [13, с. 42].

Во-первых, обратите внимание на последнее толкование "*informō*" — "мыслю"! Во-вторых, в этом же словаре отмечается, что в русском языке слово "информация" встречалось еще во времена Петра I, однако следует отметить, что сколько-нибудь массового распространения этого слова не было. Это подтверждается его отсутствием в Толковом словаре В. И. Даля (1866 г.), в энциклопедии Брокгауза и Ефрона (1896 г.), в первом издании Большой Советской энциклопедии (1935 г.). Не использовали слово "информация" и писатели того времени, по крайней мере, оно не встречается в произведениях А. С. Пушкина, Л. Н. Толстого.

Можно считать, что в современный научный и обыденный оборот термин "информация" ввели Н. Винер, К. Шеннон в середине 20-го столетия: "*... нам пришлось, — пишет Н. Винер, — разработать статистическую теорию количества информации. В этой теории за единицу количества информации принять "количество информации, передаваемое при одном выборе между равновероятными альтернативами"*" [14, с. 55].

Н. Винер применял термин "информация" к описанию разных сущностей:

1. Под информацией понимается "количество информации" — мера на пространстве случайных событий.

2. Г. Н. Поваров, редактор перевода книги Н. Винера, пишет: *"Любой сигнал, любую информацию, независимо от ее конкретного содержания и назначения, можно рассматривать как некоторый выбор между двумя или более значениями, наделенными известными вероятностями, и это позволяет подойти ко всем процессам с единой меркой, с единым статистическим аппаратом"* [14, с. 17].

Информация в этом случае — не просто сигнал, а только его формальная, статистическая составляющая, в нашей формулировке — это данные.

3. Н. Винер писал: *"Наши нынешние автоматические машины отличаются тем, что они могут правильно работать лишь в том случае, если они получают от человека необходимую им информацию и в самой точной форме. Это означает, что характер информации, вводимой в машину, в общем смысле должен быть точно и заранее известен человеку. Живые организмы, наоборот, развивают необходимую им информацию благодаря постоянному взаимодействию с природой. Это означает, что возникновение информации в живых организмах есть исторически развивающийся процесс. Мне бы хотелось еще раз подчеркнуть, что речь идет именно о взаимодействии и обмене с окружающей средой. Можно сказать, что живые организмы сами себя организуют. ... Различие между человеком и машиной, прежде всего, заключается в том, что в организме человека число элементов по порядку величин во много раз больше, чем обладает машина. ... Помимо этого количественного различия, существует и качественное различие. Преимущество человека состоит в его гибкости, в его умении работать с неточными идеями. Это означает, что человек обладает фантазией, другими словами, он создает понятие"* [14, с. 303—304].

Из этой цитаты следует, что информация — есть нечто, что "развивается", "возникает", только у "живого организма", что "создает понятие", т. е. информация — это смысл, понимание чего-либо.

К. Шеннон подчеркивал: *"Основная задача связи состоит в точном или приближенном воспроизведении в некотором месте сообщения, выбранного для передачи в другом месте. Часто сообщения имеют значения, т. е. относятся к некоторой системе, имеющей определенную физическую или умозрительную сущность, или находятся в соответствии с некоторой системой. Эти семантические аспекты связи не имеют отношения к технической стороне вопроса. Существенно лишь, что посылае-*

мое сообщение является сообщением, выбранным из некоторого множества возможных сообщений" [15, с. 243—244].

Здесь необходимо обратить внимание на то, что Н. Винер назвал свою книгу "Кибернетика или управление и связь в **животном и машине**", подчеркнув, тем самым, что предлагаемая им теория применима к животным и искусственным системам, машинам, а не к естественным объектам (планеты, горы, реки, вода, химические элементы и т. д.). Современный взгляд на то, что кибернетические методы можно распространить на естественные неживые объекты вызывает большое сомнение. Полезность же использования "количества информации", как математически точной меры на пространстве случайных событий для изучения любых явлений, никто не оспаривает.

Итак, об информации заговорили в середине прошлого века. Значит ли это, что и самого явления, которое мы сегодня связываем с информацией, до середины 20 века не существовало? Нет. Известный ученый Г. Р. Громов в своей книге пишет, что исторически первый этап развития информационной технологии начался с открытия *"... человеком элементов технологии длительного хранения на материальном носителе отдельных, наиболее характерных зрительных образов, связанных с накопленными знаниями"* [16, с. 8].

Не раскрывая термин "информация", Г. Р. Громов говорит о начале информационной технологии со времени регистрации умозрительных образов ("информационных образов") на материальных носителях, т. е. со времен появления наскальной живописи (25—30 тыс. лет назад).

Это положение входит в противоречие с утверждением: *"Специалистам, хорошо знакомым с основными положениями общей теории систем, кибернетики и синергетики, с точкой зрения, ... что информация является результатом (функцией) деятельности человеческого сознания, ... трудно согласиться"* [1, с. 65].

Позволим себе отметить, что есть немало специалистов, которые думают иначе. Высказывание Г. Р. Громова приведено выше. Можно привести мнение и академика РАО А. А. Кузнецова, который отмечает: *"С современной научной точки зрения, в формировании которой информатике принадлежит ведущая роль, наиболее характерным признаком живого организма, отличающим его от тел неживой природы, является наличие в нем информационных процессов"* [17].

По поводу того, что Н. Винер связывал понятие информации с живыми объектами, мы уже писали, обратимся теперь к синергетике. Г. Хакен —

основатель синергетики, пишет: *"Поскольку во многих случаях самоорганизация возникает из хаотических состояний, сначала мы должны развить методы адекватного описания таких состояний. Очевидно, что хаотические состояния содержат в себе неопределенность. ... мы должны иметь дело с неопределенностями, или, более точно, с вероятностями. Следующий вопрос состоит в том, как описывать системы, относительно которых имеется очень мало сведений. Это ... приводит нас к основным концепциям теории информации"* [18, с. 34].

Таким образом, Г. Хакен под теорией информации понимает математическую теорию связи: *"В задачу теории информации входит нахождение меры количества информации"* [18, с. 65].

Итак, одна из сущностей, которой соответствует термин информация — это умозрительное построение, с помощью которого строится мера на пространстве случайных событий.

Вернемся к цитате из работы Г. Р. Громова, в ней первый этап информационной технологии связывается с возможностью *"длительного хранения зрительных образов, связанных с накопленными знаниями, на материальных носителях"*.

Разберемся в этом процессе подробнее, посмотрим, кто и что участвует в этом процессе. Во-первых, в процессе обязательно должен участвовать человек, который знания накопил. Во-вторых, должен участвовать человек, который эти знания получает. В-третьих, должен быть материальный носитель, с помощью которого и происходит передача знаний. И только, в-четвертых, речь может идти о возможности долговременности хранения того, что человек хотел передать.

Четвертый элемент связан с регистрацией информационных образов, считает Г. Р. Громов. Из чего следует, что отдельно существуют информационные образы и их регистрация (в частности книгопечатание), но эти разные объекты **терминологически не выделены**, что видно из утверждения Г. Р. Громова: *"Книгопечатание — первая информационная революция"* [16, с. 11].

По нашему мнению, если можно говорить об *"информационной революции"*, то обязательно следует упомянуть и *"информационную эволюцию"*.

Информационная эволюция начинается с возникновения человека, появления речи (языка). Следующим важным моментом следует считать возникновение письменности (ок. 2350 г. до н. э.), которая является не меньшей революцией, чем современное книгопечатание, появившееся спустя три тысячелетия (1445 г.).

Практически все современные подходы к определению сущности информации основываются на положениях из словаря под редакцией акаде-

мика А. П. Ершова: *"Данные в информатике — факты или идеи, выраженные средствами формальной системы, обеспечивающей возможности их хранения, обработки или передачи. Такую формальную систему называют языком представления данных; синтаксис этого языка — способом представления информации; его семантику или прагматику — информацией". Указанное соотношение терминов "данные" и "информация" рекомендовано большинством терминологических справочников, но на практике они обычно трактуются как синонимы; да и сама информатика занимается не столько информацией, сколько данными"* [19, с. 816].

Представляется, что именно синонимичность понятий "информация" и "данные", неточное представление о сущности "информации" и приводят к тому обилию разных толкований термина "информация".

Исторический анализ появления термина "информация", его современные толкования, разные представления о сущности самого явления, в котором проявляется информация, позволяют выдвинуть гипотезу: информация — это нечто, что формируется в аппарате мышления живого существа под внешним воздействием и прошлой информации. Новая информация невозможна без уже имеющейся информации, которую надо уметь хранить. Внешнее воздействие — это данные, воспринимаемые органами чувств. Часть нервной системы живого существа, которая отвечает за мышление, формирование понятий и смысла, будем называть аппаратом мышления. Данные, попавшие в аппарат мышления, инициируют (запускают) процесс формирования информации, который интерпретирует полученные данные на основе уже имеющейся информации. Можно представить себе различные ситуации, возникающие в связи с получением разных данных, рассмотрим пример.

Пешеход стоит на перекрестке, регулируемом светофором. Свет светофора с помощью органа чувств (зрения) попадает в аппарат мышления пешехода. Можно сказать: к пешеходу попали данные, которые должны будут им проинтерпретированы, должна быть образована информация, на основе которой он принимает решение — идти, или стоять. Сформированная информация зависит от знания "языка светофора". Необученные животные на сигналы светофора не реагируют, так как, несмотря на полученные данные, у них не образуется информация (понимание, смысл), соответствующая ситуации.

Нельзя считать, что информация — это некая субстанция в аппарате мышления человека, по

крайней мере, сегодняшнее состояние нейрофизиологии это не подтверждает. Скорее всего, следует считать термин "информация" синонимичным терминам "смысл", "понимание", а термин "данные" следует использовать для обозначения того, с помощью чего происходит передача этого смысла.

Подводя некоторый итог сказанному, дадим основные положения гипотезы.

1. Мир существует объективно. Микроскопическая часть мироздания связана с человеком. Насколько сегодня понятно науке, только человек имеет разум и может познавать природу, понимать законы, которым подчиняется Вселенная, и сознательно их использовать. Все, что существует во Вселенной, будем называть реальностью.

2. Реальность бывает естественная и искусственная. Естественность понимается как все то, что существует в мироздании, за исключением того, что придумал и сделал человек. Познанием законов мироздания занимается человек, который выделил естественные науки, одна из основных — физика. Искусственность понимается по Г. Саймону [20] как все то, что создано в результате деятельности человека. Человек появился в результате эволюции, и как любое животное — естественный объект. Способность к мышлению — естественное свойство, а интеллект, который создается и развивается в процессе общения человека с себе подобными, имеет искусственное происхождение, так как интеллект развивается в процессе работы человека, в том числе благодаря обучению и самообучению.

3. Сущность, в которой появляется термин "информация", — это сложное явление, связанное с формированием и передачей смысла между высокоразвитыми живыми существами.

4. Из множества высокоразвитых живых существ следует выделить человека, интеллект которого развит настолько, что он способен познавать окружающий мир, создавать **универсальные искусственные объекты**. На современном этапе развития науки следует, на наш взгляд, ограничиться изучением информации с точки зрения человека. Некоторые животные тоже имеют интеллект и могут генерировать информацию, но все дело в уровне. Уровень интеллекта человека и животных не сопоставим.

5. Человек, используя знания естественных (физических, биологических) законов и свой интеллект, создает искусственные объекты. Общественные институты — это искусственные объекты, т. е. результат работы интеллекта человека.

6. Передача информации возможна только между людьми и только опосредованным способом.

Эта передача осуществляется только за счет того, что существует язык, с помощью которого происходит образование понятий, смысла. Смысл может быть передан другому человеку с помощью слов (данных).

7. Данные не имеют смысла. Смысл появляется только тогда, когда данные попадают к человеку. Смысл появляется и существует только в совокупности "человек — слово", или точнее "интеллект — данные". Отдельно данные существуют как некая реальность, не имеющая смысла (значения). Наше представление совпадает с рассуждением известного философа М. Мамардашвили: "... книга читается и существует только тогда, когда ее читают. Другого существования она не имеет" [21, с. 151].

Точно можно передать только полностью формализованный смысл, а это есть алгоритм, но алгоритм можно передать только с автоматом его реализующим.

8. Автоматические системы создаются только человеком, в них происходит обработка только данных (сигналов) по алгоритмам, разработанным человеком, с помощью устройств, придуманных человеком.

9. В настоящее время открыто четыре вида взаимодействия любых объектов. Это так называемые фундаментальные (физические) взаимодействия: слабое, сильное, электромагнитное, гравитационное. Все остальные взаимодействия объектов (людей в том числе) имеют в своей основе перечисленные. Что происходит в аппарате мышления человека при интеллектуальных действиях, пока науке неизвестно. Очевидно лишь, что в основе всех интеллектуальных действий (получения данных, их интерпретации, освоении новых понятий и пр.) лежат материальные, физические и химические процессы.

На основе приведенных положений можно дать следующее определение информации: **Информация** — это смысл (понимание, представление, интерпретация), возникающий у человека в результате получения им данных, взаимоотношений с предшествующими знаниями и понятиями [22]. Под **данными** предлагается понимать результат физического процесса (оформленный в некотором доступном виде, например, слов в формальном алфавите, электрических сигналов), получаемый, передаваемый, обрабатываемый либо человеком, либо устройством [22]. При этом необходимо помнить, что человек принимает и обрабатывает данные с помощью органов чувств, а устройство реализует прием, передачу и обработку данных с помощью алгоритма, придуманного человеком. Теперь можно **знания** предста-

вить в виде упорядоченной, согласованной, доказываемой в некоторой системе понятий информации. Необходимо различать знания отдельного человека и возможность проверки этих знаний.

Известно, что понятие информации более четко проявляется в информационных процессах, основой которых является **информационное взаимодействие**, представляющее собой взаимодействие между людьми посредством передачи между ними данных, в результате которого происходят изменения в ощущениях, мнениях, представлениях, знаниях (в психологических терминах — в ментальном опыте).

Информационное взаимодействие, при таком его определении, невозможно в технических системах, что отличает наш подход от подхода в работе [23]. Физическое взаимодействие является необходимым, но не достаточным для информационного взаимодействия.

В результате можно говорить об информационном процессе, как о процессе, при котором информация (смысл, знание), существующая у источника, должна с помощью данных, посылаемых адресату, инициализировать у него соответствующую информацию, наиболее адекватную информации у источника.

Источник и адресат — это люди, участвующие в информационном процессе, один из этих элементов может отсутствовать. Под **интеллектуальными процессами** будем понимать два симметричных процесса, происходящих у людей: первый процесс происходит у человека, стремящегося донести информацию, имеющуюся у него, неважно в какой форме и кому, в связи с этим происходит формирование цели сообщения, смысла сообщения, формы сообщения, превращение сообщения в данные (отчуждение своего смысла); второй процесс происходит у человека, получающего данные и стремящегося понять смысл сообщения, соотнести смысл с той информацией, которая у него имеется (приватизация чужого смысла).

Возникает вопрос, что происходит между двумя устройствами, между которыми установлена связь, или, что происходит в автомате, или компьютере? В этих случаях целесообразно говорить об **информатических процессах**, заранее предусмотренных и формализованных человеком. Термин информатические процессы введен по аналогии с физическими процессами, которыми занимается физика, химическими процессами, которыми занимается химия и т. п.; так как есть наука информатика, то она занимается изучением и разработкой информатических процессов, т. е. информатические процессы — это процессы приема, хранения, обработки и передачи данных.

Весьма условно можно считать: информационный процесс = двум интеллектуальным процессам + информатический процесс.

Информационный процесс, по нашему мнению, является искусственным процессом, потому что выполняется при участии и под контролем человека.

Для нормального функционирования информационного процесса необходимо построение **информационной системы**, в которой протекает процесс. Так как информационный процесс состоит из совокупности интеллектуального и информатического процессов, то логично предположить, что и информационная система состоит из **интеллектуальной и информатической систем**.

Для работы информационной системы требуется использование соответствующих ресурсов, в том числе информационных ресурсов. Использование информационных ресурсов требует информационных технологий. Качество информационных технологий зависит от информационной культуры. Культура определяет общество, в котором мы живем. Если мы говорим об информационном обществе, то оно определяется информационной культурой. Это и есть базовые понятия, которые, в свою очередь, разделяются на соответствующие интеллектуальные и информатические составляющие.

Чем занимается информатика

Читатель должен сделать свой вывод о рассмотренных выше гипотезах и следствиях, возникающих при их принятии. Принятие первой или второй гипотезы приводит к разным пониманиям информатики.

Если справедлива гипотеза атрибутивного подхода, то тогда следует принять утверждение К. К. Коллина, что информатика *"из технической дисциплины о методах обработки данных при помощи средств информационной техники все больше превращается в фундаментальную науку об информации и методах информационного взаимодействия в природе и обществе"* [1, с. 64].

Правда, при этом следует честно признать, что информатика никогда до настоящего времени не занималась информационным взаимодействием в естественных системах (системах, в которых человек не принимает никакого участия), и ей только предстоит стать фундаментальной естественной наукой и потеснить физику. В этой связи следует обратить внимание на то, что в предыдущей цитате информатика пока признается технической дисциплиной, и пока только находится в стадии превращения в фундаментальную науку.

Конечно, более интересно заниматься фундаментальной наукой, чем одной из многих технических наук.

Если же принять вторую гипотезу, что информация — это смысл чего-либо, и что она (информация) генерируется у человека при поступлении данных, то тогда следует признать, что каждая наука занимается соответствующей информацией (смыслом) и отчуждает соответствующие данные.

Посмотрите на то, как проходят дискуссии в СМИ, на научных конференциях, в научных журналах. Создается впечатление о дефиците смысла. Задача информатики (информатических процессов) заключается в рационализации обработки и передачи данных любой науки, любой сферы деятельности вплоть до религии. Физика занимается пониманием общих законов природы, т. е. физики формируют информацию (смысл) о законах природы, затем передают данные другим людям в виде физических законов, приборов, сделанных на их основе.

Сама информатика занимается пониманием того, как следует работать с данными — это и есть информация (смысл) о том, как сохранить данные, как их обработать, и отчуждается этот смысл в виде алгоритмов.

Социальные науки занимаются пониманием (информацией) того, как функционирует общество и как сделать так, чтобы члены общества жили в благоприятной обстановке. Информатика и в социальных науках занимается обработкой данных о социальных процессах, т. е. информатическими процессами, примененными к социальным данным.

В большинстве государственных документов информация понимается как данные [24].

Дадим следующее определение: *информатика — наука, изучающая информатические процессы и разрабатывающая информатические системы, наука о формализации задач из любых предметных областей, разработке алгоритмов для их решения и методов решения этих задач с использованием компьютеров и компьютерных сетей* [22].

Информатику, как и ее прародительницу — математику, можно рассматривать с точки зрения внутреннего развития, или ее влияния на внешний мир. Если математика для специалистов других специальностей является языком или средством описания предметной области, то информатику можно рассматривать как инструмент, который используется всеми науками, всеми сферами деятельности для совершенствования своей деятельности. В грубом приближении можно сказать: **Если математика — это язык наук, то информа-**

тика — это инструмент наук и всех видов деятельности.

Это не мешает информатике, точно так же, как математике, развиваться внутри себя, как самостоятельной науки.

Остается невыясненным вопрос: какое влияние имеет информатика как учебный предмет на формирование мировоззрения у школьников и студентов?

Сегодняшнее состояние научного мировоззрения у населения страны удручающее. Определенную лепту в это состояние внесла информатология. Хотя информатика, по нашему мнению, техническая наука, она как никакая другая соприкасается и с гуманитарными науками, и со спортом и даже религией. Сегодня видно, что в информатике лучше, чем в каком-нибудь другом учебном предмете, можно показать место информационного подхода в современной науке и вообще жизни. Именно информатика, по нашему мнению, должна акцентировать внимание учащихся, что основой всех процессов есть только четыре фундаментальных взаимодействия. В эти взаимодействия вложены информационные взаимодействия, в которые вложены процессы общения, затем процессы образования и обучения. В информационных процессах главным элементом является человек, а в информатизации главными являются информатические процессы. Разделение информационных процессов на интеллектуальные (творческие) и информатические (формальные) процессы, как и любая детализация должна привести к более глубокому пониманию многих проблем современного общества.

С благодарностью рассмотрим любые замечания и мнения по поводу статьи.

Список литературы

1. **Колин К. К.** Сущность информации и философские основы информатики // Информационные технологии. 2005. № 5. С. 63—70.
2. **Архипов Г. И., Садовничий В. А., Чубариков В. Н.** Лекции по математическому анализу.: Учебник для университетов и пед. вузов / Под ред. В. А. Садовничего. М.: Высш. шк. 1999. 695 с.
3. **Философский** словарь / Под ред. И. Т. Фролова. М.: Политиздат, 1991. 560 с.
4. **Моисеев Н. Н.** Человек, среда, общество. М.: Наука, 1982. 240 с.
5. **Каломцев Б. Б.** Динамика и информация: 2-е изд. М.: Редакция журнала "Успехи физических наук", 1999. 400 с.
6. **Мигдал А.** Поиски истины. М.: Мол. гвардия, 1983. 239 с.
7. **Пенроуз Р.** Новый ум короля: О компьютерах, мышлении и законах физики: Пер. с англ. М.: Едиториал УРСС, 2005. 400 с.
8. **Колин К. К.** Социальная информатика: Учебное пособие для вузов. М.: Академический Проект; М.: Фонд "Мир", 2003. 432 с.
9. **Колин К. К., Трошин Е. В.** Критика некоторых методологических подходов в информатике и информационное образование // Открытое образование. 2005. № 2. С. 81—89.

10. **Ожегов С. И., Шведова Н. Ю.** Толковый словарь русского языка: 80 000 слов и фразеологических выражений. 4-е изд., доп. М.: Азбуковник, 1999. 944 с.

11. **Юзвшин И. И.** Основы информатиологии: Учебник. М.: Международное издательство "Информатиология"; "Высшая школа", 2000. 517 с.

12. **Кругляков Э. П.** Лженаука. Чем она угрожает науке и обществу? // Доклад на Президиуме РАН 27.05.03. <http://www.humanism.al.ru/>

13. **Черных П. Я.** Историко-этимологический словарь современного русского языка. Т. I. 7-е изд., стереотипное. М.: Рус. яз. — Медиа, 2006. 622 с.

14. **Винер Н.** Кибернетика или управление и связь в животном и машине. 2-е изд. М.: Советское радио, 1968. 328 с.

15. **Шеннон К.** Работы по теории информации и кибернетике: Пер. с англ. М.: Иностранная литература, 1963. 830 с.

16. **Громов Г. Р.** Очерки информационной технологии. М.: ИнфоАрт, 1993. 336 с.

17. **Кузнецов А. А.** О концепции содержания образовательной области "информатика" в 12-летней школе // Информатика и образование. 2000. № 7. С. 2—7.

18. **Хакен Г.** Синергетика. М.: Мир, 1980. 404 с.

19. **Математический** энциклопедический словарь. М.: Советская энциклопедия, 1988. 847 с.

20. **Саймон Г.** Науки об искусственном: Пер. с англ. М.: Мир, 1972. 147 с.

21. **Мамардашвили М. К.** Как я понимаю философию. М.: Издательская группа "Прогресс", "Культура", 1992. 416 с., "Мысль в культуре". С. 143—154.

22. **Фридланд А. Я.** Информатика: процессы, системы, ресурсы. М.: БИНОМ. Лаборатория знаний, 2003. 232 с.

23. **Кузнецов Н. А.** Информационное взаимодействие в технических и живых системах // Информационные процессы. Т. 1. 2001. № 1. С. 1—9. <http://www.jip.ru/2001/1-1-2201.htm>.

24. **Федеральный закон РФ** от 27 июля 2006 г. № 149-ФЗ "Об информации, информационных технологиях и о защите информации".

Уважаемые читатели!

Публикацией данной статьи А. Я. Фридланда редакция продолжает дискуссию, начатую в № 5, 2005 г. журнала. Напоминаем, что оценки и суждения, высказываемые участниками дискуссии, могут не совпадать с точкой зрения редакции.

Редакция журнала «Информационные технологии»

Поздравляем юбиляра!



Заведующему кафедрой «Интеллектуальные технологии и системы» Московского государственного института радиотехники, электроники и автоматики (технического университета), академику Международной академии информатизации, кандидату технических наук, профессору

Валентину Викторовичу НЕЧАЕВУ,

известному ученому в области систем автоматизации управления и обработки информации, члену редколлегии журнала «Информационные технологии» исполнилось **70** лет.

Талантливый ученый и блестящий преподаватель Валентин Викторович является автором концепции моделирования сложных систем — конфигурационного моделирования, руководителем одной из ведущих отечественных научных школ в области искусственного интеллекта, одним из организаторов принципиально нового направления подготовки специалистов в области искусственного интеллекта со специализацией «Интеллектуальные информационные технологии, ресурсы и системы». Под его руководством выросла большая плеяда молодых инженерно-технических и научно-педагогических кадров. Высокие человеческие качества в сочетании с высоким профессионализмом, принципиальностью и ответственностью снискали ему большое уважение учеников и коллег.

*Дорогой Валентин Викторович!
Поздравляем Вас с юбилеем и желаем Вам крепкого здоровья, большого счастья,
новых творческих успехов в Вашей многогранной деятельности!*

Редколлегия и редакция журнала.

ИНФОРМАЦИЯ

МЕЖДУНАРОДНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ "ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СИСТЕМ"

МАЙОРКА, ИСПАНИЯ

15—25 сентября 2008 года

ОРГАНИЗАТОРЫ КОНФЕРЕНЦИИ

Российская академия наук, Отделение информационных технологий и вычислительных систем РАН (ОИТВС РАН), Академия инженерных наук РФ, Национальная академия наук Украины, Министерство образования и науки РФ, Центр информационных технологий в проектировании РАН (ЦИТП РАН), Федеральное агентство по информационным технологиям Минсвязи РФ (ФАИТ), Институт радиотехники и электроники РАН (ИРЭ РАН), Международный центр по информатике и электронике (Интер ЭВМ), Московский государственный технический университет им. Н. Э. Баумана.

ОРГКОМИТЕТ

Козлов Валерий Васильевич —

председатель оргкомитета,
академик,

вице-президент РАН, директор Математического института им. В. А. Стеклова РАН
(г. Москва)

Гуляев Юрий Васильевич —

сопредседатель,
академик

Директор ИРЭ РАН
(г. Москва)

Гридин Владимир Николаевич —

зам. председателя, д-р техн. наук, профессор, директор ЦИТП РАН (г. Москва)

На конференции будут рассматриваться следующие вопросы:

- Применение информационных технологий в проектировании объектов новой техники и исследовании систем сложной физической структуры;
- Инфотелекоммуникационные технологии;
- Математическое и информационное обеспечение современных проблемно-ориентированных САПР;
- Информационные технологии создания беспроводных энергетических систем;
- Программирование искусственного интеллекта в приложениях;
- Современные технологии в микро-, опто-, кванто-, нано-, акусто- и магнитоэлектронных системах;
- Математическое моделирование технических систем;
- Стохастическое моделирование и прогнозирование;
- Математическое моделирование живых систем;
- Международное научно-техническое сотрудничество.

По всем вопросам, связанным с участием в конференции,
обращаться к техническому секретарю конференции
тел.: (495) 596-44-27,
тел/факс (495)599-63-77,
e-mail: info@ditc.ras.ru

Научно-техническая конференция (с международным участием)

ИНФО-2008
"Инновации в условиях развития информационно-коммуникационных технологий"

г. Сочи, 1—10 октября 2008 г.

Председатель программного комитета проф. Домрачев В. Г., председатель оргкомитета проф. Увайсов С. У.

Цель конференции: обсуждение инновационных решений в технике, образовании, экономике, социальной сфере и других областях на основе развития ИКТ. Предполагается обсуждение достижений в решении проблем качества, безопасности, диагностики, преобразования информации, корпоративного образования, информационных технологий в обучении, правовых аспектов интеллектуальной собственности и др.

Секции:

- Основные направления развития ИКТ, стимулирующие инновационные решения.
- Инновационные достижения в конкретных предметных областях (число и название секций определяются в зависимости от тематики представленных докладов).

Формы участия в работе конференции:

- пленарный доклад (до 25 мин);
- секционный доклад (до 10 мин);
- стендовый доклад;
- в качестве слушателя.

Основные сроки и условия участия в конференции:

Для участников, выступающих с докладами, отправка материалов докладов и заявок на участие — до 15 июня 2008 г.

Размер организационного взноса для участников конференции составляет 1400 руб.

Адрес оргкомитета:

109028, г. Москва, Б. Трехсвятительский пер., д. 1—3/12, стр. 8, МИЭМ, каф. РТУиС, Увайсову С. У., Долматову А. В.

<http://www.diag.ru/conf>, E-mail: conf@diag.ru,

тел.: (495) 776-64-53, (926) 744-10-14, (926) 383-07-40 (с 10.00 до 22.00 по моск. вр.)

CONTENTS

Adzhalov V. I., Tarasov V. Yu. <i>Methods for Servicing Voice Calls on the Internet</i>	2
Shackov V. G., Nopin S. V. <i>The Using of IP-telephony in Save Mode</i>	7
Svechnikov S. V. <i>Relevant Search and Automatic Classification of Network Resources</i>	11
Chesnavsky A. A. <i>Semantic Change Detection of Web-Sites</i>	16
Zhusov D. L. <i>Algorithm of Detection the Computer Attacks on the Web-Server</i>	23
Mistrov L. E. <i>Method of an Estimation of Effectiveness of Application of Complexes of Information Individual and Group Safety of Organizational-Engineering Systems in Conflict Indeterminacy</i>	26
Ageev L. D., Kalashnikov V. S. <i>The Computed Design of Flat Quasi Log-Periodic Antennas Conductive Plot</i>	31
Plotkin D. A. <i>New Method of Image Compression Based on JPEG Baseline and Binary Interval Transformation Method</i>	34
Pekunov V. V. <i>The Automatization of Parallel Programming in Simulation of Multiphase Media</i>	37
Cheremisinov D. I. <i>Parallel Program Development for Combinatorial Problem of Logic Syntheses</i>	42
Knyazev E. G., Shopyrin D. G. <i>Automatic Classification of Source Code Changes by Means of Multidimensional Statistical Analysis</i>	48
Kulakov S. M., Trofimov V. B., Bondar N. F., Chaban S. V. <i>To the Development of Intelligent System of Defects Cognition of Rolled Iron Surface</i>	53
Alguliev R. M., Abdullaev R. S. <i>Analyses of the Ways of Reducing Traffic Jams in the Urban Infrastructure</i>	59
Mukhanov L. E. <i>Fraud Detection System in the Payment Card Field</i>	62
Drogin A. V. <i>Designing Working Procedures and Software for Budgeting Subsystem of the Automated Enterprise Management System</i>	66
Sinitzyn S. V., Khlytchiev O. I. <i>Enterprise Processes Transformation to Main Objects of Docflow System for Program Project</i>	70
Fridland A. Ja. <i>About Essence of the Information: Two Approaches</i>	75

Адрес редакции:

107076, Москва, Стромьинский пер., 4/1

Телефон редакции журнала **(495) 269-5510**

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Художник *В.Н. Погорелов*.
Технический редактор *О. А. Ефремова*. Корректор *Е. В. Комиссарова*

Сдано в набор 06.03.2008. Подписано в печать 21.04.2008. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.
Усл. печ. л. 10,78. Уч.-изд. л. 12,06. Заказ 358. 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