# Информатика и её применения

Том 7 Выпуск 3 Год 2013

## СОДЕРЖАНИЕ

Подход к автоматизированному контролю работы системы извлечения данных с веб-сайтов
***А. М. Андреев, Д. В. Березкин, И. А. Козлов, К. В. Симаков*** **2**

Построение новостного рекомендательного сервиса реального времени с использованием NoSQL СУБД
***П. А. Клеменков*** **14**

Верифицируемое отображение модели данных, основанной на многомерных массивах, в объектную модель данных
***С. А. Ступников*** **22**

Исследование графа категорий английской версии Википедии
***А. В. Шкотин*** **35**

Методы активной аутентификации на основе анализа динамики работы пользователей с клавиатурой
***В. Ю. Каганов, А. К. Королёв, М. Н. Крылов, И. В. Машечкин, М. И. Петровский*** **40**

Проблемы сетевого доступа к научным журналам
***А. В. Глушановский, Н. Е. Калёнов*** **56**

Моделирование систем поддержки принятия решений синергетическим искусственным интеллектом
***И. А. Кириков, А. В. Колесников, С. В. Листопад*** **62**

Семантика аспектно-ориентированного моделирования данных и процессов
***С. П. Ковалёв*** **70**

Когнитивная интероперабельность экспертного взаимодействия в задаче обработки русско-французских параллельных текстов: лингвокогнитивные аспекты
***О. С. Кожунова*** **81**

Разработка имитационной модели сбора и обработки данных экспериментов на ускорительном комплексе НИКА
***В. В. Кореньков, А. В. Нечаевский, В. В. Трофимов*** **94**

Оценки скорости сходимости распределений некоторых случайных сумм к устойчивым законам
***В. Ю. Королев, Л. М. Закс*** **102**

Универсальный метрический тезаурус русского языка
***Л. А. Кузнецов, В. Ф. Кузнецова, А. В. Капнин*** **106**

Approximation of a multidimensional dependency based on linear expansion in a dictionary of parametric functions
***M. G. Belyaev and E. V. Bunaev*** **114**

Abstracts **126**

Об авторах **130**

About Authors **132**

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 2–13

# ПОДХОД К АВТОМАТИЗИРОВАННОМУ КОНТРОЛЮ РАБОТЫ СИСТЕМЫ ИЗВЛЕЧЕНИЯ ДАННЫХ С ВЕБ-САЙТОВ*

А. М. Андреев[1], Д. В. Березкин[2], И. А. Козлов[3], К. В. Симаков[4]

**Аннотация:** Системы извлечения данных с веб-сайтов используют информацию о разметке HTML-страниц. Для обеспечения бесперебойной работы таких систем необходимо решить проблему своевременного обнаружения изменений структуры веб-сайтов. В статье предложен подход к решению этой проблемы, предполагающий наличие двух этапов детектирования изменений верстки: оперативного и отложенного. В основе первого из них лежит кластеризация, при этом HTML-документ рассматривается как вектор некоторых характеристик. Второй этап основан на сравнении распределений этих характеристик для эталонного и тестового наборов документов. Проведена экспериментальная оценка предложенного подхода, демонстрирующая его практическую применимость.

**Ключевые слова:** сбор текстовой информации; парсинг веб-сайтов; кластеризация; статистический анализ HTML-верстки

## 1 Введение

При разработке промышленных систем интеллектуальной обработки текстов класса Text Mining приходится сталкиваться с задачами сбора текстовой информации из открытых интернет-источников, ее унификации и накопления. Методы автоматической обработки текстов (кластеризация, полнотекстовый поиск, выявление скрытых зависимостей) могут эффективно использоваться лишь при наличии актуальной, регулярно пополняющейся базы документов.

В данной статье рассматривается решение задачи качественного сбора информации с новостных веб-сайтов. Эта информация включает в себя текст новости, а также сопутствующие метаданные: название, дату публикации, автора новости и др. Под качественным сбором в первую очередь подразумевается очистка текста новости от окружающей его служебной информации: меню сайта, рекламных баннеров, блоков социальных сетей, комментариев пользователей и т. д.

Основное внимание в данной работе уделено проблеме своевременного обнаружения изменения структуры опрашиваемых веб-сайтов. Предлагаемый подход может быть использован как для обработки новостных сайтов, так и для сбора сообщений из электронных библиотек, блогов, форумов и социальных сетей.

## 2 Постановка задачи

### 2.1 Функционирование системы сбора

Существует множество подходов к организации сбора открытых текстовых материалов с веб-сайтов. Как правило, система сбора использует информацию об HTML-разметке целевых страниц для поиска в них нужной информации [1]. Эта информация используется правилами распознавания, записываемыми на принятом в системе формальном языке.

Распространение получили как ручной способ описания правил, когда правила распознавания формирует программист [2], так и автоматизированный способ, когда правила формируются автоматически на основе обучающей выборки, подготовленной оператором [3–5]. Имея набор правил, система сбора выполняет периодический опрос веб-сайтов в поисках новых материалов.

В данной работе рассматривается система, выполняющая сбор информации на основе правил, заданных вручную программистом. Основные функциональные элементы системы сбора представлены на рис. 1.

В рамках системы осуществляется периодический опрос открытых интернет-источников и получение новых текстовых материалов. Система сбора выполняет чтение RSS (Rich Site Summary) (или HTML) ленты сайта, откуда извлекаются

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL-2012).

[1]Московский государственный технический университет им. Н. Э. Баумана, arkandreev@gmail.com
[2]Московский государственный технический университет им. Н. Э. Баумана, dmitryb2007@yandex.ru
[3]Московский государственный технический университет им. Н. Э. Баумана, kozlovilya89@gmail.com
[4]Московский государственный технический университет им. Н. Э. Баумана, skv@ixlab.ru

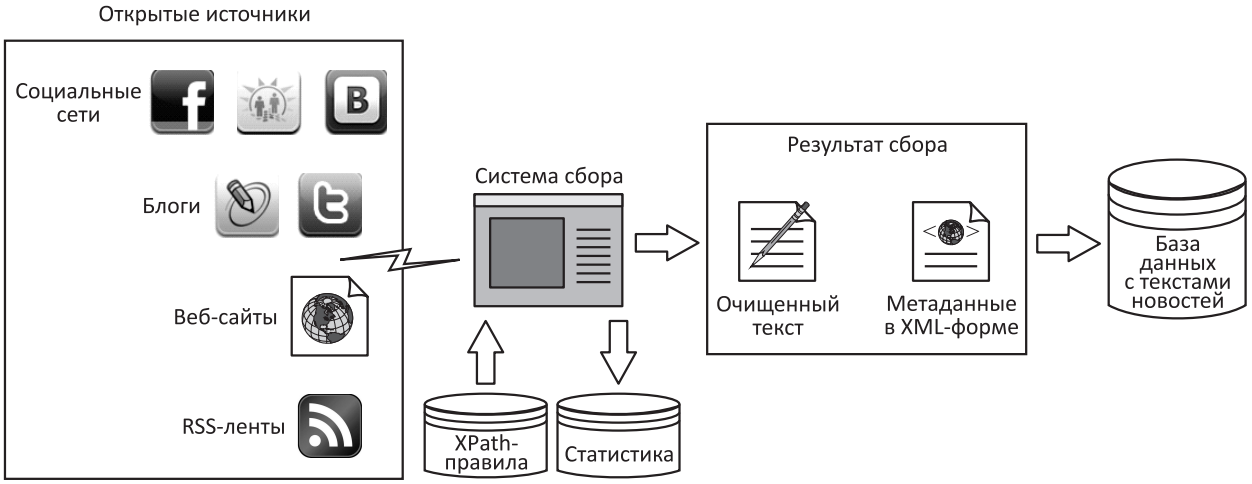



**Рис. 1** Функционирование системы сбора

метаданные о каждом документе: название, аннотация, время публикации и URL текста. Далее по полученному URL (Uniform Research Locator) осуществляется чтение страницы с текстом документа, выполняется построение объектной модели (DOM — Document Object Model) этой страницы, откуда и выполняется извлечение чистого текста на основе имеющихся XPath-правил. Результат сбора представляет собой чистый текст документа и XML-файл с метаданными. Далее эта информация заносится в базу данных, где осуществляется ее накопление и аналитическая обработка. Кроме этого, система выполняет постоянную регистрацию и накопление статистической информации о состоянии и структуре опрашиваемого веб-сайта.

## 2.2 Задача обнаружения сбоев

Все методы сбора информации с веб-сайтов, использующих особенности разметки страниц, объединяет то, что при изменении верстки сайта возникает необходимость перенастраивать правила распознавания. При выполнении круглосуточного опроса целевых сайтов своевременность обнаружения изменения верстки является весьма актуальной задачей, поскольку система сбора фактически перестает работать до тех пор, пока оператор не откорректирует набор правил распознавания.

В простейшем случае при существенном изменении структуры сайта система сбора станет выдавать в качестве результата пустые текстовые документы. Однако существуют достаточно сложные ситуации, когда при изменении верстки система сбора начинает извлекать тексты не полностью либо фрагменты из других участков сайта, например комментарии пользователей. Именно выявлению таких нетривиальных ситуаций посвящена данная статья.

## 2.3 Существующие подходы к решению задачи

В работах, посвященных теме выявления сбоев систем извлечения данных [6–8], представлено несколько подходов к решению вышеуказанной задачи. Большинство из них основано на оценке статистических характеристик документов, извлекаемых системой. При этом оценке может подвергаться как отдельно взятый документ [6] (в этом случае вычисляется вероятность его корректности, которая затем сравнивается с задаваемым пользователем пороговым значением), так и их набор [8] (оценке подвергается схожесть законов распределения случайных величин, соответствующих характеристикам документов из обучающей и тестовой выборок. Для сравнения используется критерий согласия Пирсона [9]).

В [8] также представлен подход, основанный на использовании методов machine learning для обучения системы обнаружения сбоев на наборах корректных документов для последующего определения правильности ее работы на новых данных. В качестве таких методов используется, в частности, одноклассовая классификация (выявление аномалий) [10].

Кроме того, анализ и выявление изменений в процессе сбора информации осуществляется на основе статистических данных и логов, накопленных системой сбора, а потому рассматриваемая задача может быть отнесена к направлению Process

Mining [11]. Эта дисциплина, находящаяся на стыке Data Mining и моделирования процессов, предлагает ряд подходов к анализу процессов на основе знаний, извлеченных из логов событий [12].

## 3 Принцип обнаружения сбоев

Для распознавания сбоев, связанных с изменением верстки, в систему сбора встраивается подсистема, осуществляющая контроль корректности поступающих документов и выявляющая сбои в их верстке. Возможны два следующих подхода к обнаружению сбоев.

1. **Анализ одной загруженной веб-страницы.** Суть данного подхода заключается в использовании классификатора, который определяет принадлежность веб-страницы к классу корректных или некорректных страниц. В своей работе классификатор использует набор выделяемых из веб-страницы признаков. Обучение классификатора осуществляется на предопределенных наборах документов обоих классов. Преимуществом такого подхода является высокая скорость реакции детектора на сбой: «плохой» документ будет выявлен непосредственно после его поступления. Однако этот метод имеет и серьезный недостаток. Документы, подвергающиеся анализу, могут сильно отличаться друг от друга. Так, иногда на вход детектора поступают «хорошие», но нетипичные для данного источника веб-страницы. Если подобных документов не было в обучающей выборке классификатора, они не могут быть корректно распознаны, и в результате происходит ложное срабатывание. При накапливании корректных документов и увеличении обучающей выборки частота возникновения таких ошибок постепенно уменьшается, но они продолжают периодически возникать.
2. **Анализ контрольной серии из нескольких последних загруженных веб-страниц.** Данный подход позволяет избавиться от ложных срабатываний. Даже если в контрольную серию попало несколько подозрительных документов, то усредненные характеристики этой коллекции останутся близкими к характеристикам эталонной обучающей выборки. Если же сомнительные документы будут поступать от источника регулярно, то через некоторое время, когда в контрольной серии их будет накоплено достаточное количество, они будут составлять значительную долю анализируемого набора. В результате характеристики контрольной серии изменятся и можно будет обнаружить сбой. Такой подход к фиксации сбоев более надежен. Причем качество проверки будет возрастать с увеличением числа документов в контрольной серии. Но это приведет к возникновению значительной задержки между моментом, в который произошел сбой, и временем его обнаружения.

Предложенный в данной работе метод сочетает преимущества двух вышеописанных подходов (рис. 2): быструю реакцию на сбой и высокое качество проверки.

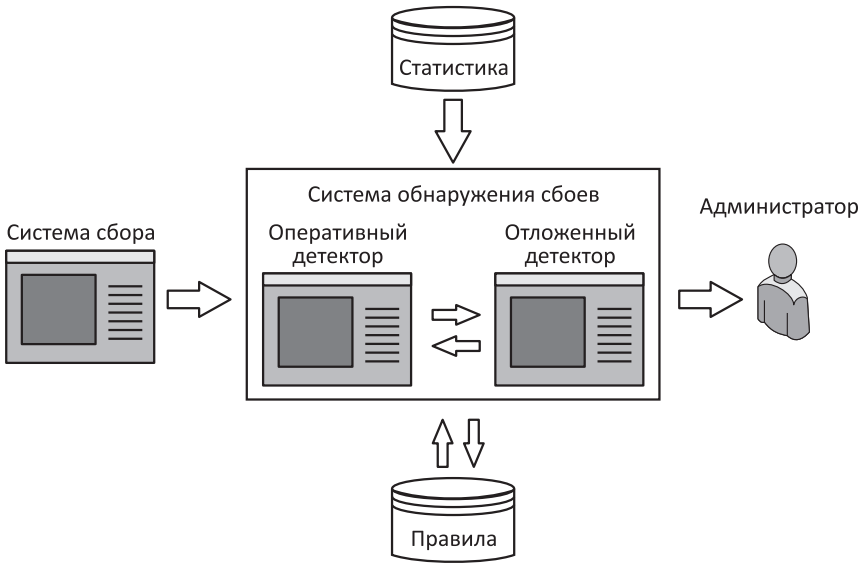



**Рис. 2** Предложенный подход

Данный метод положен в основу подсистемы контроля корректности загружаемых документов. Подсистема представляет собой двухступенчатый детектор сбоев. Один из его компонентов — «оперативный детектор» — проверяет документы непосредственно в момент их поступления и делает предварительный вывод о вероятности сбоя. Если вероятность высока, выполняется проверка «отложенным детектором», уточняющая этот результат.

# 4 Предложенные модели документов

В основе системы обнаружения сбоев лежит модель анализируемых данных. Два основных компонента системы работают с разными входными данными и анализируют различные характеристики, поэтому для каждого из них предложена своя модель: модель документа, подвергающаяся обработке «оперативным детектором», и модель набора документов, анализируемая «отложенным детектором».

## 4.1 Модель документа

Под моделью документа понимается совокупность его характеристик, учитываемых «оперативным детектором» при его обработке.

При создании детектора для системы сбора выбор параметров производился с учетом некоторых особенностей функционирования системы. Текст на целевых веб-страницах обычно разбит на параграфы (HTML-элемент $\langle p \rangle$). Также внутри текстовых параграфов могут встречаться стилевые элементы разметки. С учетом этих факторов для оценки корректности документов были выбраны следующие характеристики:

- объем веб-страницы, содержащей статью ($P$);
- суммарный размер параграфов документа ($S$). Учитывается только текст, без HTML-элементов;
- число параграфов в статье ($N$);
- дисперсия размера параграфа в рамках документа ($V$);
- количество HTML-элементов различных типов, включенных в текст документа. Для сокращения типов HTML-элементов они были сгруппированы по нескольким категориям. Были выделены классы наиболее часто встречающихся элементов: «Гиперссылки ($H$)» (в этот класс попал элемент href), «Текстовые блоки ($B$)» (br, div, span), «Форматирование текста ($S$)» ($i$, $b$, $u$, em, strong), «Изображения ($I$)» (img). Остальные теги попали в класс «Прочее ($O$)». Для каждой категории был введен параметр (соответственно $T_H$, $T_B$, $T_S$, $T_I$ и $T_O$), значение которого равно числу элементов соответствующего класса, включенных в текст документа.

Таким образом, каждый документ характеризуется рядом параметров (в данном случае — девятью), поэтому с точки зрения детектора документ представлен девятимерным случайным вектором, элементами которого являются значения перечисленных характеристик:

$$X = (P, S, N, V, T_H, T_B, T_S, T_I, T_O)\,. \qquad (1)$$

## 4.2 Модель набора документов

Для описания модели набора из нескольких документов заметим следующее. Группы характеристик $(P, S, N, V)$ и $(T_H, T_B, T_S, T_I, T_O)$ имеют разную природу. Характеристики первой группы описывают свойства текста документа, тогда как характеристики второй группы отражают свойства его разметки. Для описания свойств набора из нескольких документов будем рассматривать эти группы характеристик отдельно.

Случайные величины группы $(P, S, N, V)$ имеют разнородные области значений. Так, величина $N$ обычно принимает значения в диапазоне от 1 до 100, величина $V$ непрерывна, а значения дискретной величины $P$ могут достигать $10^5$. В связи с этим для последующего анализа удобно все величины привести к дискретному виду, а области их значений отобразить на множество фиксированной мощности. Для этого необходимо разбить область значений каждой величины группы $(P, S, N, V)$ на фиксированное количество интервалов равной длины. Число таких интервалов $m$ выбирается в зависимости от объема выборки. Одним из наиболее распространенных способов определения оптимального числа интервалов является формула Стерджесса $m = 1 + \log_2 n$, где $n$ — количество документов в наборе [13].

Для снижения вычислительной сложности алгоритмов, использующих предлагаемую модель, в контексте набора из нескольких документов будем рассматривать величины $(P, S, N, V)$ независимо друг от друга. Поэтому с точки зрения величин $(P, S, N, V)$ модель для набора документов будет представлять собой следующие четыре статистических ряда:

$$\left.\begin{array}{ll} P^n = (P_1, \ldots, P_m)\,; & S^n = (S_1, \ldots, S_m)\,; \\ N^n = (N_1, \ldots, N_m)\,; & V^n = (V_1, \ldots, V_m)\,, \end{array}\right\} \qquad (2)$$

где $P_i$, $S_i$, $N_i$ и $V_i$ — частота попадания в $i$-й интервал значения величин $P$, $S$, $N$ и $V$ соответственно на выборке из $n$ документов.

Для учета в модели (2) величин ($T_H$, $T_B$, $T_S$, $T_I, T_O$) рассмотрим другой подход к представлению информации о HTML-элементах. В $i$-м документе выборки встречается определенное количество тегов каждой из выделенных ранее пяти категорий $H$, $B$, $S$, $I$ и $O$. Обозначим эти количества $T_H^i$, $T_B^i$, $T_S^i$, $T_I^i$, $T_O^i$ соответственно. Просуммируем их по всем документам выборки и получим следующие значения:

$$T_H = \sum_{i=1}^{n} T_H^i\,;\;\; T_B = \sum_{i=1}^{n} T_B^i\,;\;\; T_S = \sum_{i=1}^{n} T_S^i\,;$$

$$T_I = \sum_{i=1}^{n} T_I^i\,;\;\; T_O = \sum_{i=1}^{n} T_O^i\,,$$

которые образуют пятиэлементный статистический ряд $T^n = (T_H, T_B, T_S, T_I, T_O)$. Этот ряд будем рассматривать в качестве модели набора документов с точки зрения частоты встречаемости в нем тегов из пяти выделенных категорий.

Таким образом, модель набора документов представляет собой совокупность следующих пяти статистических рядов:

$$\left.\begin{gathered} P^n = (P_1, \ldots, P_m)\,;\;\; S^n = (S_1, \ldots, S_m)\,; \\ N^n = (N_1, \ldots, N_m)\,;\;\; V^n = (V_1, \ldots, V_m)\,; \\ T^n = (T_H, T_B, T_S, T_I, T_O)\ . \end{gathered}\right\} \qquad (3)$$

# 5 Оперативный детектор

## 5.1 Принцип работы оперативного детектора

Быстродействующий компонент детектирующей системы представляет собой бинарный классификатор, который на основании значений параметров документа делает вывод о его корректности или некорректности. При выборе метода классификации нужно учитывать, что при обучении оперативного детектора в большинстве случаев количество «хороших» документов намного больше числа «плохих». В некоторых случаях в обучающей выборке может вообще не содержаться некорректных документов. Поэтому было решено проводить обучение классификатора на позитивных примерах, но при этом его работа была организована следующим образом: в режиме проверки документов детектор должен считать корректными лишь статьи, похожие на элементы обучающей выборки. Определим эту схожесть в терминах выбранной модели документа.

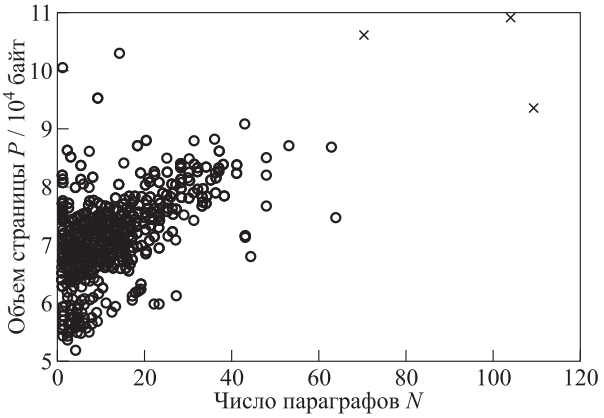



**Рис. 3** Распределение значений параметров $N$ и $P$

Каждый документ представлен девятимерным вектором. Для примера рассмотрим двумерную проекцию множества таких векторов, соответствующего набору новостей с сайта kp.ru, на плоскость, задаваемую параметрами $N$ (количество параграфов в статье) и $P$ (объем веб-страницы, содержащей статью) (рис. 3).

Точки не распределены в пространстве равномерно, они сгруппированы в некоторых областях. Новый документ, поступающий на проверку, можно считать корректным, если соответствующая ему точка попадает в одну из таких областей. Если же точка находится в отдалении от этих зон (как, например, три точки в правой верхней части рисунка, помеченные крестиком), то соответствующая статья является подозрительной.

Таким образом, обучение оперативного детектора сводится к выделению таких областей, а классификация статей на корректные и некорректные — к определению, попадает ли документ в одну из выделенных областей.

Рисунок 3 демонстрирует применение предложенного подхода для определения корректности объектов с двумерными векторами характеристик, но аналогичным образом может осуществляться классификация и в случае большей размерности векторов. Однако с ростом размерности для формирования плотных областей требуется существенно увеличивать обучающую выборку. Учитывая предполагаемые объемы наборов документов (десятки тысяч), при использовании девяти характеристик добиться высокой плотности при сохранении небольшого количества выделяемых зон невозможно.

Таким образом, описанная в (1) модель документа в виде 9-мерного вектора оказывается неудобной для непосредственного использования оперативным детектором, поэтому в нее были внесены изменения. Заменим девятимерный вектор $X$ на на-

бор векторов меньшей размерности $(Y_1, Y_2, \ldots, Y_k)$, каждый из которых содержит некоторое подмножество элементов $X$. Будем выбирать этот набор векторов исходя из следующих соображений:

(1) нужно по возможности использовать векторы наименьшей размерности (двумерные) для получения максимальной плотности кластеров;

(2) нужно избегать использования векторов, которые могут оказаться бесполезными для некоторых источников.

Второй пункт относится прежде всего к характеристикам, отражающим количество HTML-элементов, включенных в текст документа. Сайты обычно применяют для оформления текста лишь небольшой набор тегов, при этом некоторые группы HTML-элементов могут не использоваться вовсе. Поэтому только некоторые из параметров ($T_H$, $T_B$, $T_S$, $T_I$, $T_O$) будут принимать ненулевые значения. Каждый сайт использует собственный подход к оформлению и выбору набора тегов, что не позволяет определить универсальный критерий полезности каждой из этих характеристик и их совокупностей. Поэтому было решено все перечисленные величины включить в пятимерный вектор $Y_1$.

Каждый из оставшихся четырех параметров является важной характеристикой структуры документа, поэтому в качестве элементов остальных векторов использовались все попарные сочетания величин $P$, $S$, $N$ и $V$. Так были получены 6 двумерных векторов $Y_2, \ldots, Y_7$.

Таким образом, модель документа, подвергающаяся обработке «оперативным детектором», представляет собой совокупность из следующих семи случайных векторов:

$$\left.\begin{array}{c} Y_1 = (T_H, T_B, T_S, T_I, T_O)\,;\;\; Y_2 = (P, S)\,; \\ Y_3 = (P, N)\,;\;\; Y_4 = (P, V)\,;\;\; Y_5 = (S, N)\,; \\ Y_6 = (S, V)\,;\;\; Y_7 = (N, V)\,. \end{array}\right\} \quad (4)$$

## 5.2 Кластеризация документов

Выделение областей необходимо производить таким образом, чтобы максимально облегчить последующую проверку принадлежности точек этим областям. Поэтому нет смысла выбирать зоны сложной формы — более эффективным решением является нахождение плотных групп точек и построение простых ограничивающих поверхностей для этих групп. Для разбиения всего множества документов из обучающей выборки на группы нужно решить задачу кластеризации. Существует множество подходов к кластерному анализу, и применение различных алгоритмов к одним и тем же входным данным может дать совершенно разные результаты [14, 15]. Основным требованием, определяющим пригодность метода для кластеризации документов, является простая, гиперсферическая форма кластеров, позволяющая получить с помощью простых ограничивающих поверхностей плотные области без разреженных участков. Среди популярных методов кластеризации (k-means [16, 17], иерархические методы [18]) наилучшим образом отвечает требованиям к виду формируемых кластеров иерархический метод средней связи [19]. Однако он имеет серьезный недостаток, характерный для всех иерархических методов — высокую вычислительную сложность (O(n2)). Тем не менее в данной работе за основу был взят этот метод, в который были внесены следующие модификации.

Ограничим число элементов, подвергающихся кластеризации методом средней связи, числом $n$. Тогда кластеризация $N$ элементов ($N > n$) будет осуществляться следующим образом.

1. Выбрать из множества документов $n$ элементов.
2. Произвести кластеризацию этих элементов методом средней связи.
3. Найти центроиды кластеров.
4. Поместить центроиды в множество точек в качестве новых элементов.
5. Повторять п. 1–4 пока в множестве не останется необходимое число элементов.
6. Определить принадлежность исходных элементов найденным кластерам.

Результат кластеризации, произведенной описанным способом при $n = 20$, приведен на рис. 4.

В качестве ограничивающих поверхностей для областей рассматривались гиперпараллелепипед, гиперсфера и гиперэллипсоид. Выбор был сделан

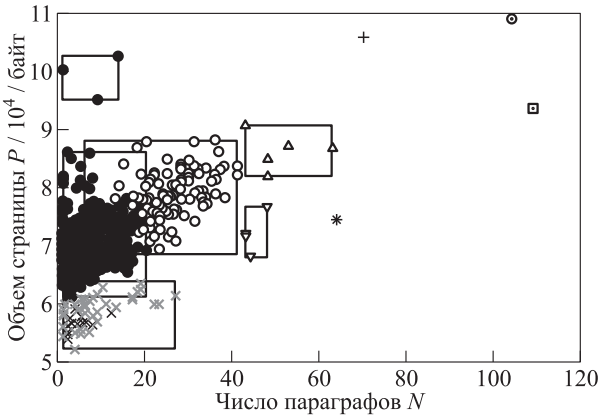



**Рис. 4** Ограничивающие поверхности кластеров

в пользу наиболее простых в построении гиперпараллелепипедов, показавших хорошие результаты при оценке плотности точек. Таким образом, каждый кластер задается набором пар $(z_{\min}, z_{\max})$, определяющих граничные значения соответствующего гиперпараллелепипеда по параметру $Z$. Элемент принадлежит кластеру, если для каждого параметра $Z$ выполняется $z_{\min} \le z \le z_{\max}$, где $z$ — значение параметра $Z$ для рассматриваемого элемента. При классификации документ считается подозрительным, если он не попадает ни в один из кластеров.

Кластеризация и построение ограничивающих поверхностей и последующая классификация загружаемых документов производятся отдельно для каждого из семи выделенных векторов (4). Таким образом, результатом классификации является набор из семи двоичных значений. Решение о корректности документа принимается на основании этого набора: статья считается корректной, если она успешно прошла проверку по каждому из семи критериев.

## 6 Отложенный детектор

Второй компонент системы обнаружения сбоев осуществляет оценку набора документов. Оценка осуществляется на основе статистических рядов (3), которые можно рассматривать как приближения к функциям вероятности соответствующих случайных величин. Идея, лежащая в основе функционирования отложенного детектора, заключается в следующем: рассматриваемые случайные величины, составляющие вектор (1), подчиняются некоторым законам распределения, которые при отсутствии сбоя остаются неизменными. Изменение же верстки с высокой вероятностью повлияет на эти законы распределения. Следовательно, две разных выборки, состоящие из корректных документов, будут обладать высокой степенью сходства. Если же одна из них будет содержать «плохие» статьи, то различие между выборками будет значительно сильнее. Таким образом, задача детектора заключается в определении степени сходства проверяемой выборки и выборки, состоящей из гарантированно корректных статей, сформированной в процессе обучения (назовем ее эталонной). На основе полученного результата принимается решение о наличии/отсутствии сбоя.

Для примера рассмотрим три выборки случайной величины $S$ (суммарный размер параграфов документа), соответствующие наборам новостей с сайта lenta.ru: эталонную (*а*); тестовую выборку, состоящую из «хороших» документов (*б*) и тестовую выборку, содержащую некорректные статьи (*в*). В качестве последних использовались новости с сайта cnews.ru.

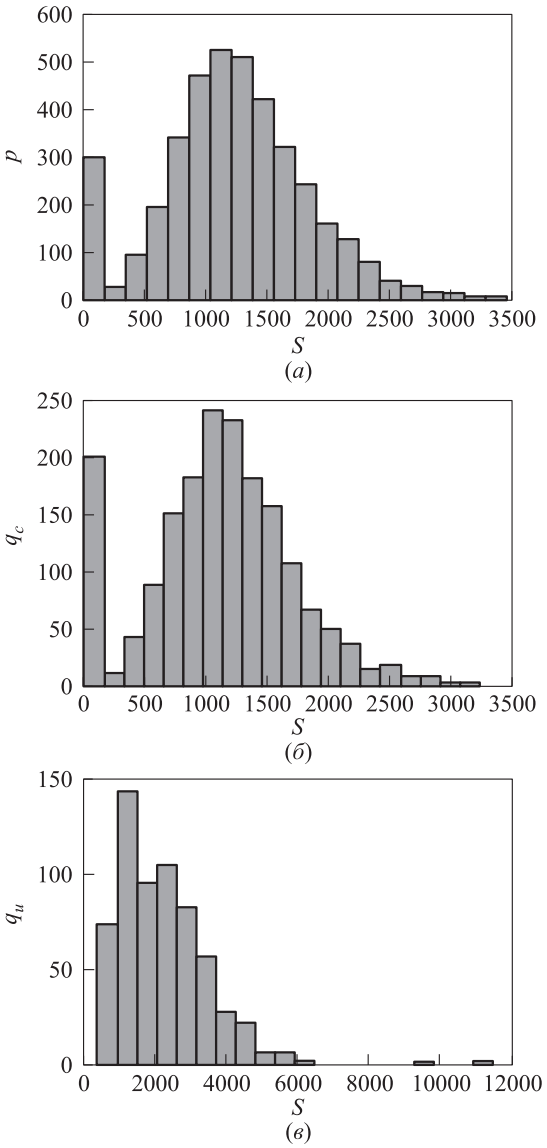



**Рис. 5** Гистограммы выборок

На рис. 5 показаны гистограммы, соответствующие этим выборкам. Первые две из них обладают высокой степенью сходства, в то время как третья значительно от них отличается.

Для оценивания сходства выборок используется относительная энтропия (расстояние Кульбака–Лейблера, KLIC [20]). Для дискретных случайных величин с функциями вероятности $p$ и $q$, принимающих значения в одном множестве $\mathcal{M} \subset \mathbb{R}$, это расстояние задается формулой

$$D_{KL}(p,q) = \sum_{x \subset \mathcal{M}} p(x) \ln \frac{p(x)}{q(x)}.$$

Вместо функций вероятности используются частоты рядов (3). При этом $p(x)$ соответствует эталонной выборке, а $q(x)$ — проверяемой.

Результатом расчета KLIC для рядов (3) являются значения $D_P$, $D_S$, $D_N$, $D_V$ и $D_N$ соответственно.

После расчета расстояния Кульбака–Лейблера встает вопрос: как по найденному значению определить, произошел сбой или нет? Необходимо задать некоторое пороговое значение $K$, такое что наличие сбоя можно определить как

$$f(D_{\mathrm{KL}}) = \begin{cases} 0\,, & D_{\mathrm{KL}} \le K (\text{сбоя нет}); \\ 1\,, & D_{\mathrm{KL}} > K (\text{произошел сбой}). \end{cases}$$

Данный порог не является фиксированной величиной, его значение зависит от числа документов в тестовой выборке. Поясним это утверждение на примере. Выберем множество $\mathbf{A} = \{A_i\}$ наборов документов $A_i$ различной мощности и вычислим для каждого из них расстояние Кульбака–Лейблера $d_i$ от эталонного закона распределения. Сопоставим натуральным числам $j$, соответствующим мощностям наборов из множества $\mathbf{A} = \{A_i\}$, числа $K_j$, определяемые как

$$K_j = \max_{A_i \in \mathbf{A}} \{D_i : |A_i| = j\}\,.$$

Рассмотрим зависимость максимального расстояния Кульбака–Лейблера от мощности набора. На рис. 6 приведена такая зависимость для новостей с kp.ru. При этом использовалась оценка характеристики $P$, отражающей объем веб-страницы, но аналогичная зависимость имеет место и для других характеристик.

Такой вид зависимости легко объясним: чем больше выборка, тем меньше на нее влияют локальные колебания значений параметров. Таким образом, при выборе порогового значения необходимо учитывать мощность анализируемого набора. Для этого необходимо определить пороговую функцию $K = h(x)$, устанавливающую соответствие между количеством документов в наборе и пороговым значением для этого набора.

Анализ рис. 6 ведет к предположению об обратно пропорциональной зависимости значения $K_j$ от $j$ и целесообразности использования аппроксимирующей функции вида $h(x) = a/x^b$. Однако проведение подобного исследования для других источников и параметров показывает, что такая функция не всегда дает приемлемый результат: в некоторых случаях зависимость имеет более сложный характер. Чтобы сделать метод определения пороговой функции пригодным для различных случаев и при этом учесть общую закономерность (постепенное уменьшение значения функции при возрастании аргумента), было решено использовать для аппроксимации функцию $h(x) = \sum\limits_{i=0}^{k} a_i/x^i$. Коэффициенты $a_i$ определяются в процессе обучения (с помощью метода наименьших квадратов (МНК) [21]), а значение $k = 7$ было выбрано на основе исследования зависимостей, характерных для различных источников. Таким образом, пороговая функция принимает вид:

$$h(x) = \sum_{i=0}^{7} \frac{a_i}{x^i}\,.$$

Для минимизации числа ложных срабатываний было решено подвергнуть функцию преобразованию, которое бы обеспечило выполнение условия $h_j \ge K_j$ для всех узлов аппроксимации. Для этого коэффициент $a_0$ необходимо увеличить на величину $\Delta = \max\limits_j (K_j - h_j)$. На рис. 7 приведены графики

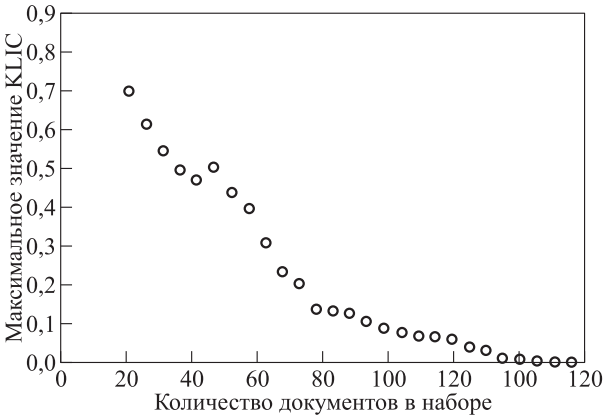



**Рис. 6** Зависимость максимального значения KLIC от мощности набора

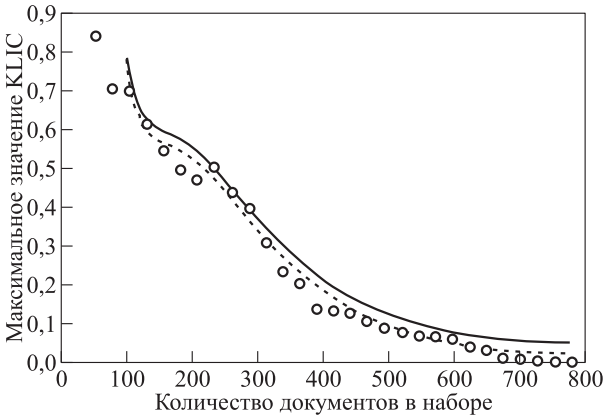



**Рис. 7** График пороговой функции

пороговой функции до (пунктирная линия) и после (сплошная линия) коррекции.

С помощью приведенной пороговой функции на основании показателей $D_P$, $D_S$, $D_N$, $D_V$ и $D_T$ получим набор из пяти двоичных значений: ($F_P$, $F_S$, $F_N$, $F_V$, $F_T$). В зависимости от количества единиц в этом наборе и от того, какие именно критерии приняли единичное значение, делается заключение о вероятности сбоя. В разработанной системе используется следующий подход:

- количество единиц в наборе равно 0 или 1 — низкая вероятность (сбоя нет);
- 2 или 3 — средняя вероятность (нельзя с уверенностью судить о наличии или отсутствии сбоя);
- 4 или 5 — высокая вероятность (произошел сбой).

## 7 Взаимодействие детекторов

Отдельной задачей является организация взаимодействия двух детекторов с целью достижения максимально эффективного функционирования системы отслеживания сбоев. Поскольку отложенный детектор осуществляет более качественный анализ и менее склонен к ложным срабатываниям, он используется для контроля работы оперативного классификатора. Этот контроль подразумевает две основные функции:

(1) проверку правильности результатов, полученных классификатором оперативного детектора;

(2) обучение классификатора. Если оперативный детектор обнаружил подозрительный документ, а отложенный детектор в результате проверки установил отсутствие сбоя, значит, произошло ложное срабатывание. Это свидетельствует о недостаточной обученности оперативного детектора. Поэтому необходимо произвести его переобучение с использованием документов, определенных им в категорию подозрительных.

В некоторых случаях ложные срабатывания могут быть обнаружены без участия отложенного детектора. Для этого оперативный классификатор был оснащен функцией самопроверки. Он способен самостоятельно отличить единичный выброс от массового поступления некорректных статей путем

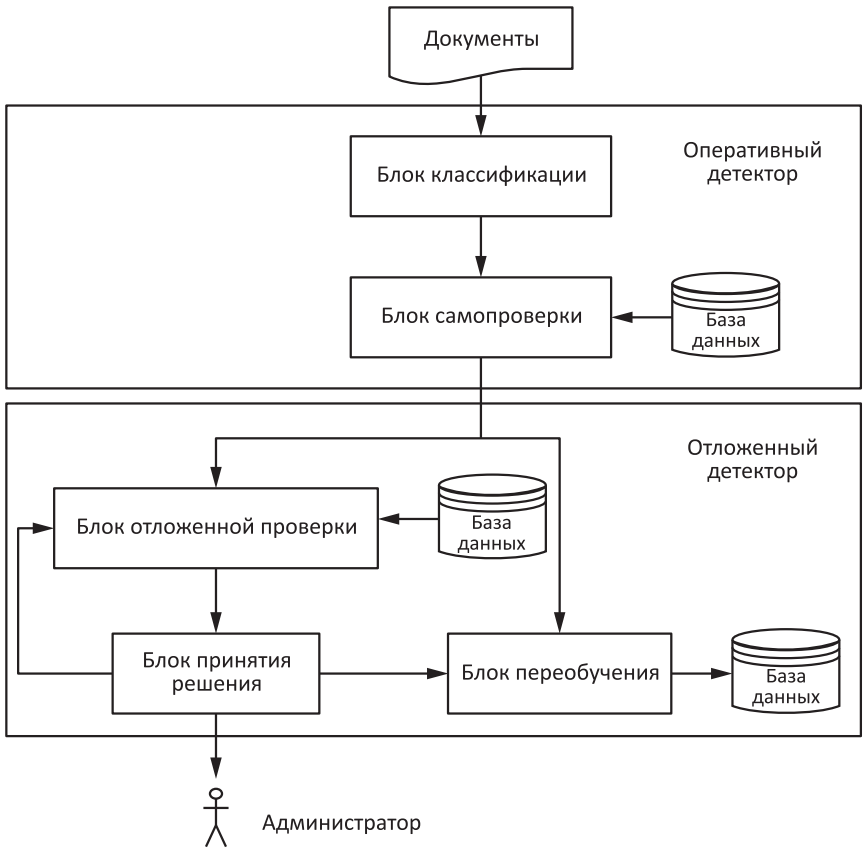



**Рис. 8** Этапы обнаружения сбоев

анализа частоты появления таких статей среди последних скачанных документов. Если эта частота меньше заданного порогового значения (например, 50%), делается вывод о ложном срабатывании и запускается переобучение. В качестве анализируемого набора при самопроверке используется группа документов, полученных в рамках последней транзакции, т. е. при последней загрузке документов с сайта.

Рассмотрим итоговый метод обнаружения изменений структуры веб-сайтов, реализованный в работе подсистемы обнаружения сбоев с учетом выбранного подхода к реализации взаимодействия детекторов. Этапы функционирования подсистемы приведены на рис. 8.

На этапе классификации оперативный детектор проверяет поступающие статьи. Документы классифицируются на корректные и подозрительные. Необходимые для классификации данные о кластерах и ограничивающих поверхностях извлекаются из базы данных.

После поступления от источника группы документов оперативный детектор выполняет самопроверку: вычисляется частота детектирования подозрительных статей в пределах текущей транзакции. Если она ниже порогового значения, но не равна нулю, делается заключение о ложном срабатывании и выполняется переход к блоку переобучения. Если частота выше порогового значения — к блоку отложенной проверки.

Работа блока отложенной проверки начинается с оповещения отложенного детектора о необходимости выполнения анализа. Выполнение проверки непосредственно после получения оповещения не имеет смысла, поскольку сбой может быть зафиксирован только после накапливания достаточного числа некорректных статей. После поступления необходимого числа документов отложенный детектор выполняет проверку этого набора. Для ее проведения из базы данных извлекаются статистические ряды эталонных выборок и коэффициенты $a_i$ пороговой функции. Результат проверки передается блоку принятия решения.

Блок принятия решения определяет дальнейшие действия подсистемы в зависимости от результата отложенной проверки. Если она показала высокую вероятность сбоя, администратор системы оповещается о необходимости корректировки системы сбора документов. Если вероятность сбоя низка, делается заключение о ложном срабатывании оперативного классификатора и выполняется переход к блоку переобучения. Если же результат анализа не позволяет с высокой долей уверенности судить о наличии или отсутствии сбоя, выполняется повторная отложенная проверка.

На этапе переобучения для оперативного детектора заново определяются кластеры и строятся ограничивающие поверхности с использованием нового, дополненного набора данных. Количество кластеров и граничные значения гиперпараллелепипедов заносятся в базу данных.

# 8 Экспериментальная проверка системы

В рамках данной работы были проведены эксперименты, направленные на анализ качества работы разработанной системы обнаружения сбоев. Эксперименты проводились на ПЭВМ со следующими основными параметрами: процессор Intel Core 2 Duo 1,8 ГГц, объем ОЗУ 2 ГБ.

Для проведения экспериментов использовалась коллекция новостей, извлеченных со следующих сайтов: mail.ru, itar-tass.com, kp.ru, rbc.ru, kommersant.ru, ria.ru, rambler.ru (табл. 1). Для обучения использовалось в общей сложности 72 888 корректных документов. При обучении оперативного детектора формировалось 10 кластеров.

При самопроверке оперативного детектора было использовано пороговое значение, равное 10%. Накопленные за время тестирования документы использовались в качестве тестовой выборки для отложенного детектора.

Целью первого эксперимента была оценка работы системы на корректных данных. В качестве входных данных использовались гарантированно корректные статьи, полученные с использованием правильных настроек системы сбора. Для проведения эксперимента использовалось в общей сложности 5169 документов.

В рамках эксперимента проверке были подвергнуты 5169 корректных статей. При первичной классификации 65 из них (1,26%) были определены как

**Таблица 1** Ложные срабатывания оперативного детектора

| Источник | $M_L$ | $M_T$ | $M_S$ | $N_D$ | $N_S$ |
|---|---|---|---|---|---|
| mail.ru | 25 296 | 2631 | 20 | 14 | 0 |
| itar-tass.com | 11 548 | 560 | 76 | 0 | 0 |
| kp.ru | 7 220 | 218 | 24 | 4 | 1 |
| rbc.ru | 3 517 | 227 | 25 | 14 | 5 |
| kommersant.ru | 5 288 | 260 | 47 | 4 | 0 |
| ria.ru | 16 519 | 1115 | 29 | 12 | 5 |
| rambler.ru | 3 500 | 158 | 15 | 17 | 13 |
| Всего | 72 888 | 5169 | 34 | 65 | 24 |

**Примечания:** $M_L$ — размер обучающей выборки; $M_T$ — размер тестовой выборки; $M_S$ — средний размер анализируемого набора документов при самопроверке; $N_D$ — количество подозрительных статей; $N_S$ — количество подозрительных статей после самопроверки.

**Таблица 2** Ложные срабатывания отложенного детектора

| Источник | $M_L$ | $M_T$ | $F_P$ | $F_S$ | $F_N$ | $F_V$ | $F_T$ | $N_F$ | $P_F$ |
|---|---|---|---|---|---|---|---|---|---|
| mail.ru | 25 296 | 2631 | 0 | 0 | 0 | 0 | 0 | 0 из 5 | $L$ |
| itar-tass.com | 11 548 | 560 | 0 | 0 | 0 | 0 | 0 | 0 из 5 | $L$ |
| kp.ru | 7 220 | 218 | 1 | 0 | 0 | 0 | 0 | 1 из 5 | $L$ |
| rbc.ru | 3 517 | 227 | 0 | 0 | 0 | 0 | 0 | 0 из 5 | $L$ |
| kommersant.ru | 5 288 | 260 | 0 | 0 | 0 | 0 | 0 | 0 из 5 | $L$ |
| ria.ru | 16 519 | 1115 | 0 | 0 | 0 | 0 | 0 | 0 из 5 | $L$ |
| rambler.ru | 3 500 | 158 | 0 | 0 | 0 | 0 | 0 | 0 из 5 | $L$ |
| Всего | 72 888 | 5169 | 1 | 0 | 0 | 0 | 0 | 1 из 35 | |

**Примечания:** $N_F$ — количество критериев, показавших наличие сбоя; $P_F$ — заключение детектора: вероятность сбоя ($L$ — низкая, $M$ — средняя, $H$ — высокая).

**Таблица 3** Оценка пропуска сбоев оперативным детектором

| Источник | $M_L$ | $M_T$ | $M_S$ | $N_D$ | $N_S$ |
|---|---|---|---|---|---|
| mail.ru | 25 296 | 356 | 25 | 356 | 356 |
| itar-tass.com | 3 500 | 356 | 25 | 356 | 356 |
| kp.ru | 11 548 | 356 | 25 | 356 | 356 |
| rbc.ru | 7 220 | 356 | 25 | 356 | 356 |
| kommersant.ru | 16 519 | 356 | 25 | 356 | 356 |
| ria.ru | 3 517 | 356 | 25 | 356 | 356 |
| rambler.ru | 5 288 | 356 | 25 | 356 | 356 |
| Всего | 72 888 | 2492 | 25 | 2492 | 2492 |

подозрительные. В результате самопроверки 41 из них был переведен в категорию корректных. Оставшиеся 24 (0,46% от общего числа) были ошибочно признаны некорректными.

Отложенный детектор показал правильный результат при проверке тестовой выборки каждого сайта (табл. 2). Ошибочное значение критерия было зафиксировано лишь в 1 случае из 35 (2,86%).

В рамках второго эксперимента (см. табл. 3) оценивалась способность системы обнаруживать сбои. Ввиду отсутствия для многих сайтов достаточного числа негативных примеров тестовые наборы были созданы искусственно: в качестве «плохих» документов использовались комментарии к новостям, полученные с сайта championat.com. Такой выбор тестовых данных обусловлен тем, что возможным последствием изменения верстки является извлечение из веб-страниц не новостей, а текстов с других участков сайта, в частности комментариев. Для проведения эксперимента использовалось 356 документов (для всех источников использовался одинаковый тестовый набор).

В рамках эксперимента проверке были подвергнуты 356 некорректных статей. При первичной классификации все они были определены как подозрительные для каждого из семи источников. В результате самопроверки никаких изменений произведено не было.

Отложенный детектор показал правильный результат для 4 источников из 7. Для оставшихся 3 источников он не смог сделать вывод о наличии или отсутствии сбоя. В 8 случаях из 35 (22,85%) значение критериев было неверным. Данным ситуациям соответствуют значения 0 соответствующего критерия в табл. 4.

Если в ходе первого эксперимента система обнаружения сбоев продемонстрировала свою работоспособность при выполнении как оперативной, так и отложенной проверки корректных данных, то с задачей обнаружения сбоев она справилась значительно хуже. Возможной причиной низкого качества работы системы при анализе некорректных документов является неудачный подход к определению результата проверки. Анализ результатов экспериментов показывает необходимость понижения порога фиксации сбоя. Кроме того, при проведении второго эксперимента критерии $F_P$, $F_S$, $F_N$, $F_V$

**Таблица 4** Оценка пропуска сбоев отложенным детектором

| Источник | $M_L$ | $M_T$ | $F_P$ | $F_S$ | $F_N$ | $F_V$ | $F_T$ | $N_F$ | $P_F$ |
|---|---|---|---|---|---|---|---|---|---|
| mail.ru | 25 296 | 356 | 1 | 1 | 1 | 0 | 0 | 3 из 5 | $M$ |
| itar-tass.com | 11 548 | 356 | 1 | 1 | 1 | 0 | 0 | 3 из 5 | $M$ |
| kp.ru | 7 220 | 356 | 1 | 0 | 1 | 0 | 1 | 3 из 5 | $M$ |
| rbc.ru | 3 517 | 356 | 1 | 1 | 1 | 0 | 1 | 4 из 5 | $H$ |
| kommersant.ru | 5 288 | 356 | 1 | 1 | 1 | 1 | 1 | 5 из 5 | $H$ |
| ria.ru | 16 519 | 356 | 1 | 0 | 1 | 1 | 1 | 4 из 5 | $H$ |
| rambler.ru | 3 500 | 356 | 1 | 1 | 1 | 1 | 1 | 5 из 5 | $H$ |
| Всего | 72 888 | 2492 | 7 | 5 | 7 | 3 | 5 | 27 из 35 | |

и $F_T$ были приняты равнозначными, однако оказалось, что некоторые из них показывают наличие сбоя значительно точнее, чем другие. Так, критерии $F_P$ и $F_N$ приняли верное значение в 7 случаях из 7, а $F_V$ — лишь в 3. Чтобы учесть различную значимость критериев, для каждого из них может быть установлен весовой коэффициент, определяющий влияние значения соответствующего критерия на результат проверки.

# 9 Заключение

В работе предложен подход к автоматизированному контролю работы системы извлечения данных с веб-сайтов. В его основе лежит двухуровневая проверка корректности веб-страниц, обеспечивающая быстроту реакции и высокое качество оценки документов.

В основе первичной классификации лежит проверка схожести документа с элементами обучающей выборки. Это позволяет системе адекватно реагировать на любые нетипичные для сайта веб-страницы. Простота выполнения такой проверки достигается с помощью предложенного метода кластеризации. Он относится к иерархическим методам, но имеет меньшую вычислительную сложность по сравнению с другими алгоритмами этого класса.

Отложенная проверка корректности основана на сравнении законов распределения. Для правильной интерпретации полученного результата используется пороговая функция, полученная путем аппроксимации МНК. Такой подход обеспечивает высокую точность проверки вне зависимости от размера оцениваемой выборки.

Проведенные эксперименты показали эффективность совместного использования двух детекторов. Предложенный подход был реализован в виде подсистемы отслеживания сбоев в системе сбора новостной информации. Данная система успешно внедрена в Совете Федерации Федерального Собрания РФ в рамках комплекса «Обзор СМИ», решающего задачу сбора, накопления и классификации новостей общественно-политической тематики.

# Литература


1. *Nikovski D., Esenther A., Baba A.* Semi-supervised information extraction from variable-length web-page lists // ICEIS 2009: 11th Conference (International) on Enterprise Information Systems Proceedings. — Milan, Italy, 2009. P. 261–266.
2. *Oro E., Ruffolo M., Staab S.* SXPath — Extending XPath towards spatial querying on web documents // VLDB Endowment Proceedings, 2011. Vol. 4. No. 2. P. 129–140.
3. *Chidlovskii B., Ragetli J., de Rijke M.* Wrapper generation by reversible grammar induction // Machine learning — ECML 2000: 11th European Conference on Machine Learning Proceedings (Barcelona, 2000). Lecture notes in computer sci. ser. Vol. 1810. — Springer, 2000. P. 96–108.
4. *Kushmerick N.* Wrapper induction: Efficiency and expressiveness // Artificial Intelligence, 2000. No. 118. P. 15–68.
5. *Tobias A.* XPath-Wrapper Induction by generalizing tree traversal patterns // Workshopwoche der GI-Fachgruppen/Arbeitskreise. — GI-Fachgruppen ABIS, AKKD, FGML, 2005. P. 126–133.
6. *Kushmerick N., Weld D. S., Doorenbos R. B.* Wrapper induction for information extraction // IJCAI 97: 15th Joint Conference (International) on Artificial Intelligence Proceedings. — Nagoya, Japan, 1997. Vol. 1. P. 729–737.
7. *Kushmerick N.* Wrapper verification // World Wide Web J., 2000. Vol. 3. No. 2. P. 79–94.
8. *Lerman K., Minton S, Knoblock C.* Wrapper maintenance: A machine learning approach // J. Artificial Intelligence Research, 2003. Vol. 18. P. 149–181.
9. *Кендалл М., Стьюарт А.* Статистические выводы и связи. — М.: Наука, 1973.
10. *Kriegel H.-P., Kröger P., Zimek A.* Outlier detection techniques // PAKDD 2009: 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining Proceedings. — Bangkok, Thailand, 2009.
11. Process Mining. http://www.processmining.org.
12. *Van der Aalst W. M. P.* Process mining: Discovery, conformance and enhancement of business processes. — Springer-Verlag, 2011.
13. *Sturges H.* The choice of a class-interval // J. Amer. Statistical Association, 1926. Vol. 21. No. 153. P. 65–66.
14. *Дюран Б., Оделл П.* Кластерный анализ. — М.: Статистика, 1977. 128 с.
15. *Мандель И. Д.* Кластерный анализ. — М.: Финансы и статистика, 1988. 176 с.
16. *Jain A., Dubs R.* Clustering methods and algorithms. — Prentice-Hall, 1988.
17. *Андреев А. М., Березкин Д. В., Морозов В. В., Симаков К. В.* Метод кластеризации документов текстовых коллекций и синтеза аннотаций кластеров // Электронные библиотеки: перспективные методы и технологии, электронные коллекции (RCDL'2008): Труды 10-й Всеросс. научной конф. — Дубна, 2008. С. 220–229.
18. *Жамбю М.* Иерархический класстер-анализ и соответствия. — М.: Финансы и статистика, 1988. 342 с.
19. *Бериков В. Б., Лбов Г. С.* Современные тенденции в кластерном анализе. — Новосибирск: Институт математики им. С. Л. Соболева, 2008. 26 с.
20. *Kullback S., Leibler R. A.* On information and sufficiency // The Annals of Math. Stat., 1951. Vol. 22. No. 1. P. 79–86.
21. Аппроксимация методом наименьших квадратов (МНК). http://alglib.sources.ru/interpolation/linearleastsquares.php.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 14–21

# ПОСТРОЕНИЕ НОВОСТНОГО РЕКОМЕНДАТЕЛЬНОГО СЕРВИСА РЕАЛЬНОГО ВРЕМЕНИ С ИСПОЛЬЗОВАНИЕМ NoSQL СУБД*

П. А. Клеменков[1]

**Аннотация:** Обсуждаются вопросы анализа взаимодействия пользователя с веб-приложением, методы проведения подобного анализа и их недостатки. Приведена реализация новостного рекомендательного сервиса с использованием существующих подходов. Описан новый подход к построению рекомендательных систем, работающих в режиме, близком к режиму реального времени, с использованием NoSQL (not only structured query language) системы управления базами данных (СУБД).

**Ключевые слова:** рекомендательные системы; minhash; mapreduce; NoSQL

## 1 Введение

Основным отличием приложений Веб 2.0 от их более старых аналогов является анализ взаимодействия пользователя с приложением и использование результатов этого анализа для модификации контента и его представления. Темпы развития сети Интернет диктуют создателям современных веб-приложений необходимость очень быстро адаптировать контент под предпочтения пользователей. Наиболее востребованным решением этой задачи стали рекомендательные системы, способные анализировать поведение пользователя, его склонности и предлагать наиболее интересное наполнение. Проблема с подобными системами заключается в том, что они недостаточно быстро реагируют на постоянно изменяющийся поток входных данных. Особенно подвержены этому новостные ресурсы. Такое поведение связано не столько с алгоритмами, применяемыми для выявления пользовательских предпочтений, сколько с архитектурными особенностями той инфраструктуры и библиотек, которые широко используются для подобного анализа. В данной статье представлен подход к организации новостного рекомендательного сервиса, призванного максимально устранить задержки в пересчете рекомендаций и обеспечить работу в режиме, близком к режиму реального времени.

## 2 Методы веб-анализа

Сегодня для анализа взаимодействия пользователя с веб-приложением применяются два основных подхода:

(1) аналитика в реальном времени;

(2) отложенная пакетная обработка логов доступа к веб-приложению.

У каждого из этих подходов есть преимущества и недостатки, на которых стоит остановиться подробнее.

### 2.1 Аналитика в реальном времени

Суть подхода заключается в том, что в ответ на взаимодействие пользователя с веб-приложением специально установленный фрагмент кода (счетчик) генерирует определенные события, обрабатываемые приложением-анализатором в реальном времени. Очевидно, что основным преимуществом подобной парадигмы является немедленное получение результатов и их обновление. Однако методы, применяемые при анализе данных в реальном времени, наиболее подходят для различных статистических расчетов (CTR, Churn Rate). При этом целые классы приложений не могут быть реализованы предложенными средствами.

### 2.2 Отложенная пакетная обработка логов доступа к веб-приложению

Этот подход строится на сборе логов доступа к веб-приложению и их последующей обработке большими частями. Имея срез данных о взаимодействии пользователей с приложением за определенный период, возможно строить сложные модели поведения и применять их, например, для выдачи рекомендаций. Современные фреймворки (например, Apache Hadoop) обеспечивают высокую

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL-2012).

[1]Московский государственный университет им. М. В. Ломоносова, parser@cs.msu.su

производительность, реализуя потоковую обработку больших объемов данных с использованием метода параллельных вычислений MapReduce [1, 2].

# 3 Рекомендательный сервис проекта Рамблер-новости

Рекомендательный сервис проекта Рамблер-новости основывается на объединении пользователей в группы по схожести интересов и вычислении наиболее популярных среди групп новостей в заданном временном окне.

## 3.1 Реализация сервиса

Суть алгоритма заключается в том, что все пользователи идентифицируются уникальными идентификаторами. Эти идентификаторы связываются с каждым HTTP-запросом к новостным ресурсам (если, конечно, запрос содержал заголовок Cookie с корректным значением). Таким образом, поведение пользователя на сайте характеризуется подмножеством логов доступа к веб-серверам. Подсчитав схожесть каждого подмножества со всеми другими, можно объединить пользователей в группы с похожими предпочтениями.

В качестве меры схожести множеств естественно использовать коэффициент Жаккарда. Однако проблема заключается в том, что время работы алгоритма подсчета этого коэффициента на нескольких миллионах множеств с сотнями и тысячами элементов является неприемлемо большим. В качестве оптимизации широко применяется вероятностный алгоритм MinHash [3]. Основная идея этого алгоритма заключается в вычислении вероятности равенства минимальных значений хеш-функций элементов множеств. Очевидно, что чем больше одинаковых элементов в двух сравниваемых множествах, тем выше указанная вероятность. А так как вычисление сигнатуры множества (минимумов используемых хеш-функций) происходит только один раз, а размер сигнатуры фиксирован, то вычисли-

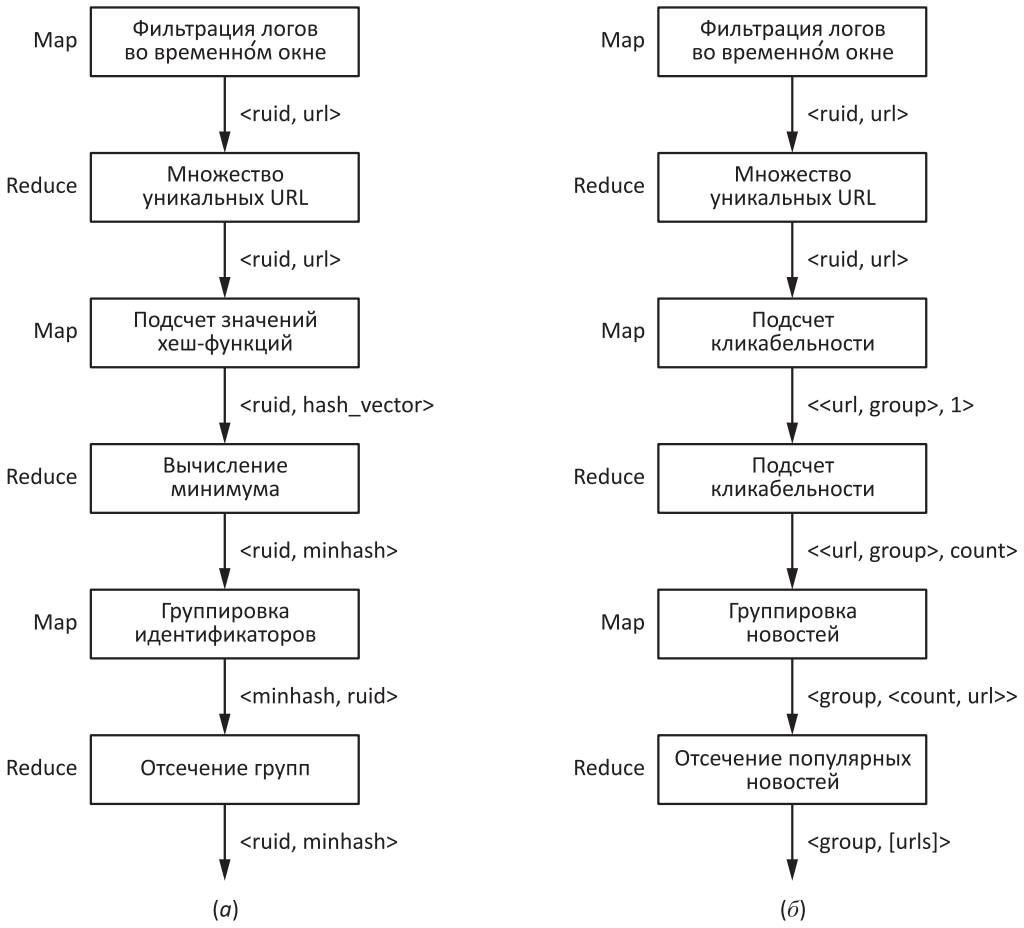



**Рис. 1** Схема работы Hadoop-реализации на первом (*а*) и на втором (*б*) этапах

тельная сложность решаемой задачи резко снижается.

Для вычисления новостных рекомендаций было принято решение производить обработку логов доступа веб-серверов Рамблер-новостей во временно́м окне 5 дней. Средний объем логов за указанный период составляет примерно 7 ГБ. Для реализации алгоритма был выбран фреймворк Hadoop, являющийся де-факто стандартом для потоковой обработки больших объемов данных.

Алгоритм вычисления рекомендаций был реализован в виде последовательности MapReduce-задач, разделенных на два этапа: подсчет групп пользователей во временно́м окне 5 сут. и подсчет рекомендаций для групп во временно́м окне 5 ч. Первый этап составляют следующие ступени (рис. 1, *а*).

1. Фильтрация логов во временно́м окне 5 сут. и генерация множества уникальных URL (uniform resource locator) для каждого идентификатора пользователя.
2. Подсчет значений хеш-функций для всех уникальных URL каждого пользователя и вычисление минимума, который становится идентификатором группы.
3. Подсчет численности групп и отсечение $\sim 100$ групп с наибольшей численностью. Необходимо заметить, что порог отсечения вычисляется статистически, поэтому имеет место небольшая дисперсия числа групп. Однако на производительность и поведение алгоритма это влияет незначительно.

Также необходимо отметить, что первоначальная реализация алгоритма имела еще один шаг, который позволял строго отсечь необходимое число групп, но ради сокращения вычислений им было решено пренебречь.

Второй этап разделен на следующие ступени (рис. 1, *б*).

1. Фильтрация логов во временно́м окне 5 ч и генерация множества уникальных URL для каждого идентификатора пользователя.
2. Подсчет кликабельности новостей во всех группах.
3. Отсечение заданного числа наиболее популярных новостей в каждой группе.

Получающиеся в результате отображения идентификаторов в группы и групп в популярные новости загружаются в хранилище Redis, позволяющее запрашивать список рекомендаций для данного пользователя в реальном времени.

## 3.2 Производительность

Приведенная реализация алгоритма использовалась в продуктивном окружении проекта Рамблер-новости более полугода, показывая приемлемое время работы. На Hadoop-кластере из 8 узлов первая ступень обсчитывалась примерно 7 мин, а вторая — 3,5–4 мин при условии, что другие задачи не выполнялись параллельно. Необходимо отметить, что важным фактором производительности MapReduce-задач является верный выбор количества мапперов и редьюсеров. Выбор количества мапперов производился автоматически. Экспериментальным путем было выяснено, что оптимальное число редьюсеров в данной конфигурации — 16.

## 3.3 Проблемы

Внимательно изучив получившуюся архитектуру и приняв во внимание проблемы, возникшие при реализации рекомендательного сервиса, можно отметить следующие аспекты.

1. Загрузка логов в HDFS (Hadoop Distributed File System) и их обработка — две не связанные задачи. В данном случае синхронизация логов выполнялась с помощью утилиты rsync, а вычисление разности между файлами в директории синхронизации и файлами в HDFS, а также загрузка новых файлов — с помощью специально написанного Makefile и shell-скриптов.
2. В Hadoop отсутствует возможность получать данные из разных источников. В частности, результаты работы первого этапа алгоритма приходилось передавать в окружение второй ступени второго этапа в виде файла в кеше Hadoop. Вследствие того что этот файл может иметь весьма внушительный размер, MapReduce-задачи на всех узлах могут столкнуться с проблемой нехватки памяти.
3. Задачи подсчета рекомендаций и их использования также не являются связанными и выполняются разными инструментами. В данном случае — Hadoop и Redis.
4. Ну и самое главное — пакетный потоковый режим работы Hadoop не позволяет хоть сколько-нибудь приблизиться к реальному времени пересчета результатов.

Отсюда возникает вопрос: можно ли решить все вышеперечисленные проблемы, воспользовавшись другим подходом? В следующей части статьи будет описана архитектура подобного решения с применением NoSQL-хранилищ данных.

# 4 Введение в NoSQL

Термин NoSQL впервые был использован в 1998 г. для описания реляционной базы данных, не использовавшей SQL. Он был вновь подхвачен в 2009 г. и использован на конференциях приверженцами нереляционных баз данных. Основной движущей силой развития NoSQL-хранилищ стали веб-стартапы, для которых важнейшей задачей является поддержание постоянной высокой пропускной способности хранилища при неограниченном увеличении объема данных. Рассмотрим основные особенности NoSQL-подхода, делающие его таким привлекательным для высоконагруженных веб-проектов [4, 5].

1. **Исключение излишнего усложнения.** Реляционные базы данных выполняют огромное множество различных функций и обеспечивают строгую консистентность данных. Однако для многих приложений подобный набор функций, а также удовлетворение требованиям ACID (atomicity, consistency, isolation, durability) являются излишними.
2. **Высокая пропускная способность.** Многие NoSQL-решения обеспечивают гораздо более высокую пропускную способность данных, нежели традиционные СУБД. Например, колоночное хранилище Hypertable, реализующее подход Google Bigtable, позволяет поисковому движку Zvent сохранять около миллиарда записей в день. В качестве другого примера можно привести саму Bigtable [6], способную обработать 20 ПБ информации в день.
3. **Неограниченное горизонтальное масштабирование.** В противовес реляционным СУБД, NoSQL-решения проектируются для неограниченного горизонтального масштабирования. При этом добавление и удаление узлов в кластере никак не сказывается на работоспособности и производительности всей системы. Дополнительным преимуществом подобной архитектуры является то, что NoSQL-кластер может быть развернут на обычном аппаратном обеспечении, существенно снижая стоимость всей системы.
4. **Консистентность в угоду производительности.** При описании подхода NoSQL нельзя не упомянуть теорему CAP (consistency, availability, partition tolerance). Согласно этой теореме, многие NoSQL-хранилища реализуют доступность данных (availability) и устойчивость к разделению (partition tolerance), жертвуя консистентностью в угоду высокой производительности. И действительно, для многих классов приложений строгая консистентность данных — это то, от чего вполне можно отказаться.

# 5 Классификация NoSQL-хранилищ

На сегодняшний день создано большое число NoSQL-хранилищ. Все они основываются на четырех принципах из предыдущего раздела, но могут довольно сильно отличаться друг от друга. Многие теоретики и практики создавали свои собственные классификации, но наиболее простой и общеупотребительной можно считать систему, основанную на используемой модели данных, предложенную Риком Кейтелем (см. табл.) [7].

Классификация NoSQL-хранилищ по модели данных

| Тип | Примеры |
|---|---|
| Хранилища ключ–значение | Redis<br>Scalaris<br>Tokio Tyrant<br>Voldemort<br>Riak |
| Документно-ориентированные хранилища | SimpleDB<br>CouchDB<br>MongoDB |
| Колоночные хранилища | Bigtable<br>HBase<br>HyperTable<br>Cassandra |
| Хранилища на графах | Neo4j |

1. **Хранилища ключ–значение.** Отличительной особенностью является простая модель данных — ассоциативный массив или словарь, позволяющий работать с данными по ключу. Основная задача подобных хранилищ — максимальная производительность, поэтому никакая информации о структуре значений не сохраняется.
2. **Документно-ориентированные хранилища.** Модель данных подобных хранилищ позволяет объединять множество пар ключ–значение в абстракцию, называемую «документ». Документы могут иметь вложенную структуру и объединяться в коллекции. Однако это скорее удобный способ логического объединения, так как никакой жесткой схемы у документов нет и множества пар ключ–значение даже в рамках одной коллекции могут быть абсолютно произвольными. Работа с документами производится по ключу, однако существуют решения, позволяющие осуществлять запросы по значениям атрибутов.
3. **Колоночные хранилища.** Этот тип кажется наиболее схожим с традиционными реляционными

СУБД. Модель данных в хранилищах подобного типа подразумевает хранение значений как неинтерпретируемых байтовых массивов, адресуемых кортежами ⟨ключ строки, ключ столбца, метка времени⟩. Основой модели данных служит колонка, число колонок для одной таблицы может быть неограниченным. Колонки по ключам объединяются в семейства, обладающие определенным набором свойств.

4. **Хранилища на графах.** Подобные хранилища применяются для работы с данными, которые естественным образом представляются графами (например, социальная сеть). Модель данных состоит из вершин, ребер и свойств. Работа с данными осуществляется путем обхода графа по ребрам с заданными свойствами.

# 6 Построение рекомендательного сервиса Рамблер-новостей с помощью NoSQL

Вспоминая недостатки реализации рекомендательного сервиса на фреймворке Hadoop, можно отметить, что NoSQL-хранилища кажутся приемлемым вариантом их устранения. NoSQL-хранилища обеспечивают высокую пропускную способность данных как при чтении, так и при записи. Из этого следует, что логи доступа к веб-приложению можно записывать непосредственно в базу данных. Важно также отметить, что при использовании документно-ориентированных решений логам можно придавать произвольный вид, не создавая жесткую схему. Это позволяет решать довольно интересную задачу хранения и обработки структурированных логов. К тому же механизм выборки документов по значениям атрибутов позволяет решать множество аналитических задач.

Большинство современных NoSQL-решений реализуют парадигму вычислений MapReduce. Наряду с фундаментальным свойством горизонтального масштабирования это дает возможность переносить алгоритмы, предназначенные для фреймворков типа Hadoop, на хранилища NoSQL, получая все дополнительные преимущества.

Учитывая высокую пропускную способность операций чтения, задачи подсчета рекомендаций и их использования можно не разделять. Следовательно, обновленные рекомендации будут тут же доступны потребителям, что приближает сервис к требованиям реального времени.

Далее следовало определиться с конкретным продуктом, который можно было бы использовать для реализации сервиса. Среди документно-ориентированных баз данных первоначальный выбор пал на проект Apache CouchDB [8]. CouchDB работает с документами, представленными в формате JSON (JavaScript Object Notation). Для работы с документами предоставляется REST API (REpresentation State Transfer Application Programming Interface). Для построения запросов к документам CouchDB и их анализа применяются так называемые «представления». По сути представление является обычной MapReduce-задачей, которая может сохранять результаты выполнения в базе. Интересной особенностью модели данных CouchDB является то, что для индексации документов и представлений используются модифицированные B-деревья. Сохраняя все особенности и преимущества стандартного B-дерева, B-деревья CouchDB реализуют режим «только добавление». Это означает, что любые операции вставки, модификации и изменения записываются в конец файла, представляющего B-дерево на диске. Такая архитектура дает два основных преимущества: высокую скорость записи и возможность исполнять MapReduce-задачи только на изменившихся данных. Однако при всех своих преимуществах CouchDB не подходила для решения поставленной задачи. Во-первых, проект не поддерживает никакого языка запросов, что сильно затрудняет выборку документов по определенным критериям. Во-вторых, важным критерием выбора была поддержка ссылок на другие документы. Подобная возможность есть в CouchDB, но работает она только на этапе эмиссии документа из map-задачи. К тому же нет возможности создания ссылок на документы из других баз. В-третьих, неоптимизированное JSON-представление документов приводит к увеличению трафика между клиентом и хранилищем, чего хотелось избежать. Окончательный выбор пал на проект MongoDB [9]. Обладая всеми преимуществами CouchDB, это хранилище устраняет перечисленные недостатки и предоставляет дополнительные удобные возможности. Они будут упомянуты в следующем разделе, описывающем реализацию рекомендательного сервиса.

# 7 Реализация рекомендательного сервиса Рамблер-новости

Первая задача, которую предстояло решить, — это запись логов в базу данных MongoDB. Первым делом требовалось определить, какое число операций записи в секунду обеспечивала выбранная конфигурация. Стоит отметить, что тестовая конфигурация представляла собой кластер из

двух узлов, на каждом их которых был запущен демон mongod без репликации. На одном их хостов запускался демон mongos, обеспечивавший шардинг документов. Для определения скорости записи был разработан простой скрипт, производивший загрузку суточных логов новостей в базу MongoDB. Лог состоял из 2 770 695 записей. Среднее время записи составило 18 мин 30 с. Таким образом, средняя скорость записи в представленной конфигурации — 2496 операций/с. Шардинг документов осуществлялся по атрибуту ruid (уникальный идентификатор пользователя). Подобный результат более чем достаточен для рассматриваемого сервиса, так как среднее количество запросов в секунду к веб-сайту Рамблер-новости существенно меньше. Однако загрузка логов из ротированных лог-файлов разработчика не устраивала. Для удовлетворения требования реального времени необходимо было обеспечить загрузку логов в базу сразу после обработки запроса веб-сервером. Для этого с помощью библиотеки ZeroMQ был разработан специализированный демон, агрегировавший логи с нескольких фронт-эндов новостей в хранилище MongoDB.

Необходимо отметить, что загрузчик логов не только производил их фильтрацию, представление в формате BSON (Binary JavaScript Object Notation) и запись в базу, но и подсчет значений хеш-функций для каждого URL. Это было обусловлено двумя факторами: снижением времени вычислений и отсутствием приемлемых реализаций быстрого хеширования в языке JavaScript (на нем реализуются MapReduce задачи в MongoDB).

После того как задача загрузки логов была решена, необходимо было перенести реализацию алгоритма подсчета рекомендаций с Hadoop на MongoDB. Возвращаясь к реализации первого этапа в подразд. 3.1, можно отметить, что задачи фильтрации логов и подсчета значений хеш-функций для них реализуются загрузчиком. Поэтому оставалось перенести только подсчет минимальных значений хешей и отсечение групп с заданной численностью (рис. 2).

Стоит обратить внимание на то, что из новой реализации пропал этап отсечения групп по численности. В первоначальной реализации отсечение делалось главным образом для сокращения времени загрузки отображения ⟨идентификатор пользователя → группа⟩ в Redis. При использовании NoSQL-хранилища подобной проблемы не возникало.

Возвращаясь к цифрам, отметим, что задача подсчета минимального хеша для суточных логов (2 770 695 записей) заняла примерно 3 мин 10 с. Это не сильно отличается от времени выполнения той же задачи на Hadoop-кластере, и почему это

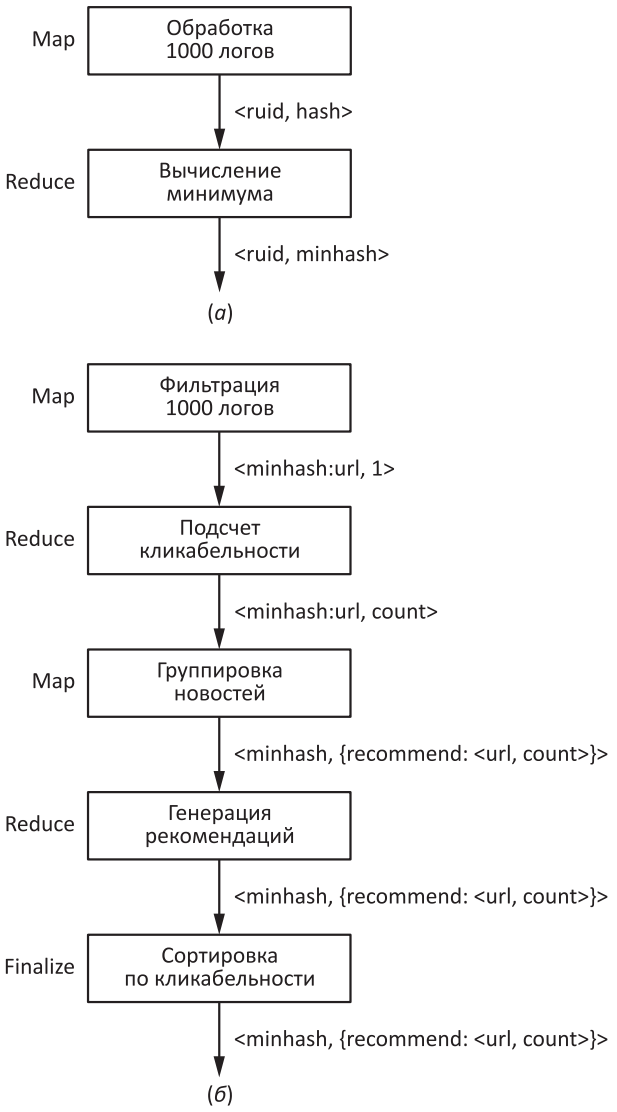



**Рис. 2** Схема работы NoSQL-реализации на первом (*а*) и на втором (*б*) этапах

происходит, вполне очевидно. Однако здесь на помощь приходит вся мощь MongoDB. Во-первых, результаты MapReduce-задач сохраняются в отдельной коллекции. Последующие вычисления можно производить только на добавленных с прошлого запуска логах, выполняя rereduce на получившихся результатах. Во-вторых, мощный язык запросов MongoDB позволяет осуществить выборку логов, добавленных с момента последнего запуска задачи. В предлагаемой архитектуре задача сохранения времени последнего выполнения и перезапуск вычислений возложена на загрузчик логов. Важно отметить, что высокая производительность библиотеки ZeroMQ позволила не масштабировать загрузчик логов, поэтому проблем с синхронизацией времени

не возникало. В-третьих, MongoDB поддерживает создание и поддержание индексов на атрибутах документов, что существенно ускоряет выборки. На основании всего вышесказанного было принято решение перезапускать задачу подсчета минимального хеша после записи одной тысячи новых логов с выборкой по атрибуту timestamp документа. Данная задача без индекса завершалась в среднем через 3 с, а с индексом — через 400–500 мс, что уже существенно приблизило разработку к требованиям реального времени.

Теперь перейдем ко второму этапу алгоритма — выработке рекомендаций (рис. 2, *б*). Здесь возникают три основные проблемы: выборка логов в заданном временно́м окне, дополнительная фильтрация и ввод данных из нескольких источников. Выборку логов в заданном временном окне можно, как и на первом этапе, осуществлять запросом по атрибуту timestamp. Стоит отметить, что MongoDB реализует capped collections. Это коллекции с заранее определенным объемом. Если объем коллекции достиг заданного порога, то новые значения затирают старые. Это интересный подход к ротации, но для рассматриваемой задачи он не подходит, так как количество логов может меняться день ото дня. Дополнительная фильтрация осуществляется регулярными выражениями JavaScript, здесь нет никаких сложностей. Проблема ввода данных из нескольких источников решается с помощью механизма DBRef MongoDB. Он позволяет создавать ссылки на связанные документы в виде вложенных документов и получать к ним доступ при выполнении map-задач. Удобная особенность DBRef следует из отсутствия схемы документов и других ограничений — ссылаться можно на несуществующие документы и коллекции. Этим фактом пользуется загрузчик логов, создавая ссылки на группы, которых еще нет.

Таким образом, первые две ступени второго этапа удалось объединить в одну: map-задача фильтрует выборку логов во временно́м окне 5 ч и возвращает пару $\langle \text{group\_id} : \text{url}, 1 \rangle$, а reduce-задача подсчитывает количество кликов по каждой новости всех групп. Среднее время выполнения этой ступени — 350 мс на той же тысяче логов. Третья ступень была просто адаптирована для исполнения MongoDB. Надо, правда, отметить, что отсечение заданного количества популярных новостей не производится. Эту задачу с целью сокращения объема вычислений было решено возложить на потребителя. Также следует сказать, что на последней ступени используется функция finalize, позволяющая видоизменить результаты reduce-задачи. В данном случае функция finalize производит сортировку новостей в группах по числу кликов.

# 8 Проблемы, возникшие при реализации сервиса рекомендаций

Естественно, при реализации сервиса возник определенный набор трудностей, о которых важно упомянуть. Первая трудность — ротирование логов. Так как в MongoDB отсутствует механизм времени жизни ключей, задачу ротирования логов приходится решать периодическим запуском отдельной MapReduce-задачи. К тому же во всех документах, требующих удаления, приходится явно хранить метку времени жизни. Вторая трудность заключается в том, что формат возвращаемых map-задачей значений должен совпадать с форматом значений, возвращаемых reduce-задачей. Из-за этого приходится создавать довольно сложные структуры, чего хотелось бы избежать. Третья трудность — это специфическое устройство шардинга в MongoDB. Ключи распределяются по узлам не равномерно, а группами. Из-за этого некоторые MapReduce-задачи на небольшом числе документов выполняются на одном узле, содержащем все ключи.

# 9 Заключение

В результате проведенного эксперимента удалось создать рекомендательный сервис, время пересчета рекомендаций в котором на каждую тысячу новых логов составляет 1,5–2 с. Для проекта Рамблер-новости подобный результат является удовлетворительным, так как 1000 новых запросов к сайту делается за чуть большее время. Стоит отметить, что алгоритм MinHash как таковой не предназначен для подсчета рекомендаций в режиме реального времени. Более того, эффективность новой реализации рекомендательного сервиса может оказаться ниже, чем предыдущая реализация с помощью фреймворка Hadoop. Однако целью данной работы было показать целесообразность применения NoSQL-подхода к построению систем анализа данных в режиме, близком к режиму реального времени. Сделанные выводы позволят реализовать на описанной платформе более подходящие рекомендательные алгоритмы, например Covisitation [5]. Важным свойством приведенной реализации является то, что задачи хранения и анализа данных удалось объединить с задачей предоставления доступа к результатам в единой системе, избежав накладных расходов на перемещение данных из одного источника в другой и улучшив общую производитель-

ность сервиса. Кроме того, предложенный подход упрощает решение повседневных задач сбора статистики о взаимодействии пользователя с веб-приложением путем анализа структурированных логов мощным языком запросов СУБД MongoDB. Можно утверждать, что применение NoSQL-подхода к решению подобного класса задач весьма перспективно и может быть использовано в продуктивном окружении высоконагруженных веб-приложений.

## Литература


1. *Dean J., Ghemawat S.* MapReduce: Simplified data processing on large clusters // OSDI'04: 6th Symposium on Operating System Design and Implementation Proceedings. — Berkeley, CA, USA: USENIX Association, 2004. P. 137–149.
2. *Venner J.* Pro Hadoop. — N.Y.: Apress, 2009.
3. *Das A. S., Datar M., Garg A., Rajaram Sh.* Google news personalization: Scalable online collaborative filtering // 16th Conference (International) on World Wide Web Proceedings, 2007. P. 271–280.
4. *Pokorny J.* NoSQL databases: A step to database scalability in web environment // 13th Conference (International) on Information Integration and Web-Based Applications and Services Proceedings, 2011. P. 278–283.
5. *Strauch C.* NoSQL databases. http://www.christof-strauch.de/nosqldbs.pdf.
6. *Chang F., Dean J., Ghemawat S., Hsieh W. C., Wallach D. A., Burrows M., Chandra T., Fikes A., Gruber R. E.* Bigtable: A distributed storage system for structured data // 7th USENIX Symposium on Operating Systems Design and Implementation Proceedings. — Berkeley, CA, USA: USENIX Association, 2006. Vol. 7. P. 205–218.
7. *Cattel R.* Scalable SQL and NoSQL data stores // ACM SIGMOD Record, 2010. Vol. 39. No. 4. P. 12–27.
8. *Anderson J. C., Lehnardt J., Slater N.* CouchDB: The definitive guide. — Sebastopol: O'Reilly Media, 2010.
9. *Chodorow K., Dirolf M.* MongoDB: The definitive guide. — Sebastopol: O'Reilly Media, 2010.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 22–34

# ВЕРИФИЦИРУЕМОЕ ОТОБРАЖЕНИЕ МОДЕЛИ ДАННЫХ, ОСНОВАННОЙ НА МНОГОМЕРНЫХ МАССИВАХ, В ОБЪЕКТНУЮ МОДЕЛЬ ДАННЫХ*

С. А. Ступников[1]

**Аннотация:** Рассматривается отображение модели данных, основанной на многомерных массивах (ММ-модели), в объектную модель данных. Изложены общие принципы отображения ММ-моделей в объектные модели данных. Рассмотрено отображение конкретной модели — Array Data Model (ADM), использующейся в системе управления базами данных (СУБД) SciDB, в язык СИНТЕЗ, использующийся в качестве канонической модели данных в технологии предметных посредников. Проиллюстрирован метод верификации отображения — доказательства сохранения информации и семантики операций при отображении. Верификация осуществляется при помощи формального языка спецификаций AMN. Практической целью работы ставилось создание базы для виртуальной или материализованной интеграции ресурсов, основанных на многомерных массивах.

**Ключевые слова:** многомерные массивы; объектная модель данных; отображение моделей данных; интеграция баз данных

## 1 Введение

Развитие науки и промышленности, широкое распространение информационных технологий ведет к накоплению огромных объемов данных как в науке, так и в бизнесе. Данные могут быть как наблюдательными, экспериментальными, так и полученными в ходе компьютерного моделирования. Данные таких масштабов (часто измеряемых уже в петабайтах) называются «большими данными» (Big Data) [1]. Они плохо поддаются обработке и анализу в рамках широко известных технологий баз данных, опирающихся в основном на реляционную модель данных.

Именно поэтому развиваются различные модели данных, нацеленные на параллельную обработку и анализ данных в распределенных средах — гридах и облаках. Важными видами таких моделей являются модели данных, основанные на многомерных массивах (array-based data models, или ADM) и называемые далее ММ-моделями. Родственны данным моделям так называемые «кубы данных», используемые в OLAP (online analytical processing) технологии [2–4]. Исследования ММ-моделей начались достаточно давно [5, 6] и продолжают развиваться. В данной статье рассматривается конкретная модель, а именно модель, используемая в СУБД SciDB [7].

История SciDB начинается с 2007 г., когда на симпозиуме по экстремально большим базам данных (XLDB — extremely large data bases) представителями науки и промышленности был сделан вывод о том, что существующие СУБД не в состоянии манипулировать объемами данных, которые появятся в ближайшем будущем. Одним из примеров поставщиков таких данных служит строящийся телескоп LSST (Large Synoptic Survey Telescope) [8]. Был также сделан вывод о необходимости разработки СУБД нового поколения, которая должна удовлетворять, в частности, следующим требованиям [9]:

- модель данных основывается на многомерных массивах, а не на кортежах;
- модель хранения базируется на версионности, а не на обновлении значений;
- масштабируемость до сотен петабайт и высокая отказоустойчивость;
- СУБД является свободно распространяемым программным обеспечением.

Некоторое время спустя был запущен международный проект под руководством Майкла Стоунбрейкера, целью которого стало создание новой СУБД, получившей название SciDB. В настоящее время свободно распространяется очередная версия системы для операционных систем (ОС) Ubuntu и RedHat.

Целью данной статьи является исследование вопроса о верифицируемом отображении ММ-моделей, и в частности ADM [10], использующейся в

*Работа выполнена при поддержке РФФИ (проект 11-07-00402-а). Статья рекомендована к публикации в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL-2012).

[1]Институт проблем информатики Российской академии наук, ssa@ipi.ac.ru

системе SciDB, в объектные модели данных для виртуальной или материализованной интеграции ресурсов при создании федеративных баз данных или хранилищ данных.

При материализованной интеграции предполагается создание хранилища данных (warehouse), в которое загружаются ресурсы, подлежащие интеграции. В процессе загрузки происходит преобразование данных из схемы ресурса в общую схему хранилища.

Виртуальная же интеграция рассматривается в статье применительно к предметным посредникам [11]. Предметные посредники представляют собой специальный вид программного обеспечения (ПО), образующий промежуточный слой между пользователем (приложением) и неоднородными информационными ресурсами. При этом данные из ресурсов не материализуются в посреднике. Федеративная схема посредника, описывающая некоторую предметную область, создается независимо от существующих ресурсов. Ресурсы, релевантные предметной области, затем регистрируются в посреднике — их схемы связываются специальными семантическими отображениями с федеративной схемой. Исполнительная среда посредников предоставляет возможность пользователям (приложениям) задавать запросы (программы) к посреднику в терминах федеративной схемы. Эти запросы переписываются в частичные запросы над информационными ресурсами, а затем исполняются на ресурсах. Результаты частичных запросов объединяются и выдаются пользователю также в терминах федеративной схемы.

Важным понятием технологии систем интеграции баз данных является каноническая модель, служащая общим языком, унифицирующим разнообразные модели ресурсов.

Необходимым предусловием интеграции ресурсов, основанных на многомерных массивах, является построение отображения соответствующей ММ-мо-дели в каноническую модель данных, сохраняющего информацию и семантику операций языка манипулирования данными (ЯМД) [12]. Это обусловлено тем, что семантические отображения, связывающие федеративную схему и схемы ресурсов, нужно проводить в единой (канонической) модели [13]. Отображение должно быть верифицируемым — доказуемо правильным.

В качестве канонической модели в данной работе рассматривается язык СИНТЕЗ [14] — комбинированная слабоструктурированная и объектная модель данных, нацеленная на разработку предметных посредников для решения задач в средах неоднородных ресурсов. Разработан прототип программных средств для поддержки среды предметных посредников с языком СИНТЕЗ в роли канонической модели [15].

С точки зрения предметных посредников СУБД, основанные на многомерных массивах, представляют собой новый вид ресурсов, подлежащих интеграции в посредниках вместе с привычными ресурсами — реляционными и объектными СУБД, веб-сервисами и т. д.

Нужно отметить, что ADM подвергается некоторой критике со стороны исследователей, продолжающих развитие моделей, основанных на многомерных массивах. Так, авторы языка SciQL [16] отмечают, что язык ADM является смесью SQL и деревьев алгебраических операций. По их мнению, язык для СУБД, основанных на многомерных массивах, должен быть интегрирован с синтаксисом и семантикой SQL:2003. Несмотря на эти замечания, модель ADM представляет несомненный практический интерес для интеграции баз данных. SciDB используется как в научных проектах, связанных с LSST (предполагается после запуска телескопа) и физикой высоких энергий, так и в коммерческих, связанных с генетикой, страхованием, финансами. Сравнительное тестирование SciDB с СУБД Postgres и статистическим ПО R показало преимущества SciDB по производительности и масштабируемости.

Статья организована следующим образом. В разд. 2 рассмотрены и проиллюстрированы основные принципы отображения модели данных ADM в язык СИНТЕЗ. Принципы обобщены на случай моделей, отличных от ADM и СИНТЕЗ. В разд. 3 рассмотрен метод доказательства сохранения информации и семантики операций при отображении моделей с использованием формального языка спецификаций AMN [17]. Метод проиллюстрирован на структурах данных и операциях ЯМД моделей SciDB и СИНТЕЗ. В разд. 4 рассмотрены некоторые родственные исследования и направления дальнейшей работы.

## 2 Отображение модели ADM в язык СИНТЕЗ

SciDB поддерживает два языка для работы с массивами: AQL (Array Query Language) и AFL (Array Functional Language). Язык AQL является SQL (Structured Query Language) подобным декларативным языком, включающим как операции языка описания данных (ЯОД), так и операции ЯМД. Язык AFL представляет собой функциональный язык манипулирования массивами, операции кото-

рого можно объединять в композиции. Допускается использование операций AFL в запросах AQL.

Операции языков и отображение будут иллюстрироваться на адаптированных примерах из сценария применения SciDB в области оптической астрономии [18], а также на простых примерах из документации SciDB [10].

## 2.1 Отображение языка определения данных

Отображение ЯОД в данном разделе описывается независимо от вида интеграции — виртуальной или материализованной.

Основной единицей определения данных в модели ADM является массив, имеющий конечное количество измерений $d_1, d_2, \ldots, d_n$ [9]. Длиной измерения называется количество упорядоченных значений в этом измерении. По умолчанию типом измерения являются 64-битные целые числа. Поддерживаются также нецелочисленные измерения, например строки или числа с плавающей точкой. Каждая комбинация значений измерений соответствует ячейке массива, которая может содержать конечное количество значений, называемых *атрибутами*. Типом атрибута может быть один из встроенных типов ADM [10].

Основная операция ЯОД ADM — создание массива — выглядит следующим образом:

```
CREATE ARRAY source
< ampExposureId: int64 NULL,
   filterId: int8,
   apMag: double >
[ ra(double), de(double), objectId=0:*];
```

Создается массив оптических источников source, измерениями которого являются координаты ra и de типа double и целочисленный идентификатор объекта. Для целочисленного измерения указаны его нижняя (0) и верхняя («*», обозначающая бесконечность) границы. Ячейка массива состоит из трех атрибутов: ampExposureId, filterId, apMag. Указано, что атрибут ampExposureId может принимать неопределенное значение NULL. В данном примере приведены только некоторые из реально используемых атрибутов и измерений.

В языке СИНТЕЗ создание массива представляется определением одноименного класса:

```
{ source; in: class;
  instance_type:{
  double ra;
  ra2long: {in: function;
            params: {-ret/long}; };
  double de;
  de2long: {in: function;
            params: {-ret/long}; };
  long objectId; metaslot lower: 0;
  higher: inf; end
  objectIdBounds: {in: invariant;
    {{all s(source(s) -> s.objectId >= 0)}}
  };
  long ampExposureId;
  short filterId;
  double apMag;
  key: { unique; { ra, de, objectId } };
  definiteness: {obligatory;
    { ra, de, objectId, filterId, apMag } };
  };
}
```

Как измерения, так и атрибуты, составляющие ячейку, представляются в языке СИНТЕЗ атрибутами типа экземпляров (instance_type) класса. Между встроенными типами ADM (int8, int64, double и др.) и встроенными типами языка СИНТЕЗ (short, long, double) устанавливается взаимно однозначное соответствие. Совокупность атрибутов, соответствующих измерениям, объявляется уникальной (инвариант key, выражаемый встроенным утверждением unique). Объявляется также, что атрибуты, соответствующие измерениям и не-NULL атрибутам ADM, должны быть определены у всех экземпляров класса (инвариант definiteness, выражаемый встроенным утверждением obligatory).

Таким образом обеспечивается сохранение отличительных свойств многомерных массивов («кубов данных»), существенным образом различающих измерения и атрибуты, составляющие ячейку:

- по набору значений измерений однозначно определяется набор значений атрибутов ячейки (уникальность измерений);
- ячейка массива всегда определяется полным набором значений измерений (определенность измерений).

Заметим также, что отсутствие в коллекции объекта с некоторым набором значений измерений означает *пустую ячейку* в массиве.

Для нецелочисленных измерений ra и de в языке СИНТЕЗ кроме атрибутов определяются функции ra2long, de2long, преобразующие нецелочисленные значения в целочисленные. Необходимость привнесения этих функций вызвана следующим. При попытке описать операции, характерные для ММ-моделей, в объектной модели (в частности, в языке СИНТЕЗ) выясняется необходимость применения принципиально различных механизмов работы с целочисленными и нецелочисленными измерени-

ями. Это вызвано различием типов измерений, возможной неравномерностью шага измерения и т. д. Для того чтобы обеспечить возможность единообразного описания операций над целочисленными и нецелочисленными измерениями и необходимы функции, приводящие нецелочисленные измерения к целочисленным.

Ограничения, связанные с нижними и верхними границами целочисленных измерений, представляются в языке СИНТЕЗ, во-первых, метаслотом соответствующего атрибута (например, objectId). В метаслоте хранится метаинформация, связанная с атрибутом как с отдельной сущностью языка. В данном случае метаслот включает два слота lower и higher, отвечающих соответственно верхней и нижней границе измерения. Во-вторых, создается инвариант (например, objectIdBounds), предикативная спецификация которого устанавливает ограничения на значения измерения для каждого из объектов класса, отвечающего массиву. Спецификация инварианта имеет вид формулы первого порядка, где all — квантор существования, «->» — импликация.

Необходимо отметить, что массив представляется в объектной модели множеством объектов класса (фактически кортежей значений атрибутов). При этом наблюдается некоторое противоречие со стремлением создателей ММ-моделей отойти от моделей, основанных на кортежах. Однако в контексте интеграции ресурсов ММ-модели это лишь один класс из большого множества разнообразных классов моделей данных. Представление специфических ММ-моделей в объектной модели является методологически и технически гораздо более простым и естественным, нежели использование многомерных массивов в качестве канонической модели.

Изложенные принципы отображения ЯОД могут быть обобщены на случай, когда канонической является объектная или объектно-реляционная модель, отличная от языка СИНТЕЗ. Также не принципиален выбор модели данных, основанной на многомерных массивах. В общем виде принципы отображения ЯОД выглядят следующим образом:

- массив отображается в коллекцию типизированных объектов (класс) объектной модели;
- измерения и атрибуты, составляющие ячейку массива, отображаются в атрибуты типа экземпляров класса;
- между встроенными типами модели, основанной на многомерных массивах, и встроенными типами объектной модели устанавливается взаимно однозначное соответствие;
- совокупность атрибутов, соответствующих измерениям, объявляется уникальной (при помощи механизма ключей, утверждений или инвариантов);
- атрибуты, соответствующие измерениям и не-NULL атрибутам ячейки массива, объявляются определенными (при помощи утверждений или инвариантов);
- для нецелочисленных измерений в типе экземпляров дополнительно определяются методы, преобразующие нецелочисленные значения в целочисленные;
- ограничения, связанные с нижними и верхними границами целочисленных измерений, отображаются при помощи инвариантов или встроенных утверждений о кардинальности соответствующих атрибутов. В случае использования инвариантов при отображении границы измерений представляются также метаданными атрибута.

## 2.2 Отображение языка манипулирования данными

При интеграции баз данных для установления семантических соотношений между схемами ресурсов и федеративной схемой необходимо отображение ЯОД исходной модели в каноническую. Язык манипулирования данными канонической модели, напротив, необходимо отображать в ЯМД исходной модели. Это связано с тем, что запросы к посреднику в канонической модели необходимо отображать в запросы к ресурсам.

Отметим отличие виртуальной и материализованной интеграции. При виртуальной интеграции отображение ЯМД обеспечивает возможность трансляции программ на языке посредника в запросы на языке ресурсов.

В случае материализованной интеграции данные извлекаются из ресурса и представляются в хранилище в канонической модели. При этом программы на языке канонической модели исполняются непосредственно на данных. Отображение ЯМД нужно лишь для того, чтобы убедиться, что отображение моделей сохраняет информацию и семантику операций. Семантически правильное отображение служит базой для процесса «извлечения–преобразования–загрузки» (ETL), формирующего из данных ресурса данные хранилища: ETL-процесс может быть выражен только в терминах канонической модели.

Язык запросов (программ) модели СИНТЕЗ представляет собой Datalog-подобный язык в объ-

ектной среде. Программа представляет собой набор конъюнктивных запросов (правил) вида

$$q(x/T) : -C_1(x_1/T_1), \ldots, C_n(x_n/T_n), (X_1, Y_1), \ldots \\ \ldots F_m(X_m, Y_m), B\,.$$

Тело запроса представляет собой конъюнкцию предикатов-коллекций, функциональных предикатов и ограничения. Здесь $C_i$ — имена коллекций (классов), $F_i$ — имена функций, $x_i$ — имена переменных, значения которых пробегают по классам, $T_i$ — типы переменных, $X_j$ и $Y_j$ — входные и выходные параметры функций, $B$ — ограничение, налагаемое на $x_i$, $X_j$, $Y_j$. Предикаты, находящиеся в голове правил, могут быть использованы в телах других правил в качестве предикатов-коллекций.

В дальнейшем будет часто использоваться запись предиката-коллекции вида source([ra, de]). Неформально это означает, что представляют интерес не объекты класса source целиком, а лишь их атрибуты ra и de. Формально запись означает сокращение от source(_/source.inst[ra, de]). Здесь знак «_» обозначает анонимную переменную, source.inst — анонимный тип экземпляров (instance) класса source, а ra и de — необходимые атрибуты типа экземпляров класса.

Будет также использоваться запись source([i, j, val1/val]), означающая переименование атрибута val в val1.

При отображении ЯМД будут сначала рассмотрены основные конструкции языка программ СИНТЕЗ, соответствующие конструкциям языка AQL. Затем будут рассмотрены конструкции СИНТЕЗ, соответствующие конструкциям языка AFL.

Начнем рассмотрение с конструкций языка СИНТЕЗ, соответствующих конструкциям языка AQL, связанных с извлечением данных.

**Предикаты-классы, условия, подзапросы.** Рассмотрим программу, извлекающую координаты (ra, de) и апертурную звездную величину (apMag) астрономических источников из класса source с условием на фильтр (filterId) и апертурную звездную величину, причем запрос q использует результаты запроса r:

```
q([ra,de,apMag]) :- r([ra,de,apMag]),
   filterId= #filterId.
r([ra,de,apMag]) :- source([ra,de,apMag]),
   apMag >= #apMag.
```

Здесь #filterId и #apMag — некоторые константы соответствующих типов.

Такая программа представляется в AQL следующим запросом:

```
SELECT apMag FROM
  ( SELECT apMag FROM source
    WHERE apMag >= #apMag )
WHERE filterId = #filterId;
```

Простые условия отображаются в AQL без изменений, рекурсивные запросы представляются вложенными запросами. Заметим, что координаты ra и de не указываются в секции SELECT — они являются измерениями и извлекаются по умолчанию.

**Соединение классов.** Соединение по определенным атрибутам (например, objectId)

```
q2([ra, de, filterId, uMag]) :-
    source([ra, de, objectId, fliterId]),
    objectSummary([objectId, uMag]).
```

представляется в AQL конструкцией JOIN-ON:

```
SELECT filterId, uMag INTO q2
FROM source
JOIN objectSummary
ON source.objectId = objectSummary.objectId;
```

где массив objectSummary имеет вид:

```
CREATE ARRAY objectSummary
<uMag: float NULL,  gMag: float NULL>
[ objectId=0:* ];
```

**Агрегация.** Рассмотрим запрос, возвращающий объекты с минимальной звездной величиной uMag:

```
q([objectId, uMag]) :-
  objectSummary(obj/[objectId, uMag]),
    uMag = min(obj.uMag).
```

Запрос представляется в AQL с использованием агрегирующей функции того же рода:

```
SELECT uMag
FROM source,
 (SELECT min(uMag) AS min FROM Source)
WHERE uMag = min;
```

**Группирование.** Рассмотрим запрос, возвращающий среднее значение звездной величины uMag, вычисленное на группе по идентификатору объекта filterId:

```
q([objectId, avgMag]) :-
    group_by({objectId},
       q2(obj/[ra,de,filterId, uMag])),
    avgMag = average(obj.uMag).
```

Здесь коллекция q2, на которой производится группирование по атрибуту objectId — результат соединения классов source и objectSummary, рассмотренных выше.

Очевидно, в AQL запрос представляется при помощи конструкции GROUP BY:

```
SELECT avg(uMag) AS avgMag
FROM q2 GROUP BY objectId;
```

Рассмотрим конструкции языка СИНТЕЗ, соответствующие конструкциям языка AQL и связанные с изменением данных.

**Обновление.** Рассмотрим запрос, изменяющий значения в квадратной матрице (см. предыдущий пример) на значения с обратным знаком в том случае, если модуль значения больше 5:

```
source(x/[i, j, val]) :-
    source(x/[i, j, val1/val]),
       abs(val) > 5, val = -val1.
```

В AQL данный запрос представляется следующим образом:

```
UPDATE source
SET val =  -val WHERE abs(val) > 5;
```

**Удаление.** Рассмотрим программу, удаляющую из базы данных класс и все его содержимое:

```
-source(x) :- source(x).;
delete_frame(source).
```

В правилах со знаком минус в голове осуществляется удаление объектов из коллекции. В данном случае из коллекции удаляются все объекты. Функция delete_frame удаляет коллекцию как объект из базы данных. Операция «;» обозначает последовательную композицию программ. В AQL данный запрос представляется при помощи операции DROP:

```
DROP ARRAY source;
```

Рассмотрим принципы отображения конструкций языка СИНТЕЗ, соответствующих конструкциям AFL, на примере расширения элементов массива в подмассивы. Каждый элемент массива расширяется в подмассив определенного размера. Значения всех ячеек подмассива дублируют значение оригинальной ячейки. Пример программы, расширяющей каждую ячейку матрицы $3 \times 3$ в подматрицу $2 \times 2$:

```
q([i,j,val]) :- {x/[i,j,val] | exists y (
  source(y/[i1/i, j1/j, val]) &
  ( i = i1*2 & j = j1*2 | i = i1*2 +1 &
  j = j1*2 | i= i1*2 &
  j= j1*2 + 1 | i= i1*2 +1 & j= j1*2 +1))}.
```

Здесь выражение $\{x/T|F(x)\}$, где $F$ — формула со свободной переменной $x$, обозначает конструкцию выделения множества; exists обозначает квантор существования.

В ADM запрос представляется с использованием операции xgrid:

```
SELECT * FROM xgrid(source, 2, 2);
```

Можно заметить, что операция AFL xgrid имеет достаточно сложно устроенный прообраз в канонической модели (это справедливо и для многих других операций). Между тем эти операции являются естественными для массивов. Поэтому при интеграции ресурсов, основанных на многомерных массивах, в канонической модели возможно использование специального класса array, инкапсулирующего специфические операции, характерные для многомерных массивов:

```
{ array; in: class;
  instance_type: {
  xgrid: { in: function;
    params: {
     +dimensions/{sequence;
      type_of_element: string;},
     +scales/{sequence;
      type_of_element: integer;}};
  };  };
}
```

В приведенном примере рассмотрена сигнатура единственной операции xgrid, параметрами которой являются последовательность имен измерений dimensions и последовательность масштабов увеличения по каждому из измерений scales. Параметром операции по умолчанию также считается класс array как коллекция объектов. При отображении ЯОД каждый класс — образ массива (например, класс source из подразд. 2.1) становится подклассом класса array:

```
{ source; in: class; superclass: array;
  instance_type: { ... };
}
```

Аналогично xgrid, операциями класса array могут быть представлены такие операции AFL, как substitute, sort, multiply и т. д.

Заметим, что решение о представлении операций, характерных для многомерных массивов, в рамках специального класса канонической модели допускает обобщение на объектные канонические модели, отличные от языка СИНТЕЗ, и модели, основанные на многомерных массивах, отличные от ADM.

Разработанные отображения ЯОД и ЯМД были частично реализованы на языке ATL (ATLAS Transformation Language) [19]. ATL-программы представляют собой декларативно-императивные трансформации, реализующие отображения произвольных исходных моделей уровня M1 (согласно

классификации MOF [20]), конформных исходной метамодели уровня M2, в целевые модели уровня M1, конформные целевой метамодели уровня M2. Модели уровня M1 являются схемами, представленными в канонической модели данных или модели ADM; модели уровня M2 есть описание абстрактного синтаксиса канонической модели или модели ADM. В качестве метамодели уровня M3, которой конформны метамодели уровня M2, рассматривается модель Ecore [21]. Синтаксис ЯОД и ЯМД ядра канонической информационной модели (языка СИНТЕЗ) и модели ADM был представлен в метамодели Ecore.

Было осуществлено построение ATL-трансформаций, реализующих отображения подмножества ЯОД модели ADM в ЯОД канонической модели и подмножества ЯМД канонической модели в ЯМД модели ADM. Подмножества ЯМД определялись конструкциями ЯОД и ЯМД канонической модели, поддерживаемыми в настоящее время в исполнительной среде предметных посредников. Поддерживаемый язык запросов канонической модели включает правила, в голове которых могут быть предикаты-коллекции, а в теле — конъюнкция предикатов-коллекций, условий соединения коллекций и других условий на значения атрибутов типов экземпляров коллекций. Условием соединения может быть только равенство атрибутов. Поддерживаются основные арифметические предикаты и функции, а также термы — вызовы функций.

# 3 Сохранение информации и семантики операций языка манипулирования данными при отображении

Метод доказательства сохранения информации и семантики операций при отображении моделей данных [22] основывается на представлении семантики моделей в формальном языке спецификаций AMN [17].

Язык AMN представляет собой теоретико-модельную нотацию, основанную на теории множеств и типизированном языке первого порядка. Спецификации AMN называются абстрактными машинами. Язык AMN позволяет интегрированно рассматривать спецификацию пространства состояний и поведения машины (определенного операциями на состояниях). В AMN формализуется специальное отношение между спецификациями — уточнение. Неформально спецификация $B$ уточняет спецификацию $A$, если пользователь может использовать $B$ вместо $A$, не замечая факта замены $A$ на $B$.

Идея метода заключается в следующем. Рассмотрим исходную модель $S$ и целевую модель $T$. Построим отображение $\theta$ модели $S$ в модель $T$ (подобно изложенному в предыдущем разделе). Выразим семантику моделей в виде абстрактных машин AMN, построив при этом машины $M_S$ и $M_T$ соответственно. При этом структуры данных моделей — классы, массивы — представляются переменными машин, различные свойства структур данных представляются инвариантами машин, характерные операции моделей данных представляются операциями машин. Рассматриваемые операции исходной и целевой модели должны быть связаны отображением ЯМД. Отображение ЯОД представляется в виде специального *склеивающего инварианта* — замкнутой формулы, связывающей состояния машин $M_S$ и $M_T$.

Будем считать отображение $\theta$ сохраняющим информацию и семантику операций, если машина $M_S$, соответствующая исходной модели, уточняет машину $M_T$, соответствующую целевой модели. Уточнение доказывается интерактивно при помощи специальных программных средств [23].

В качестве иллюстрации основных принципов выражения семантики моделей ADM и СИНТЕЗ в AMN рассмотрим частичные AMN-спецификации, соответствующие данным моделям.

Спецификация, выражающая семантику объектной модели языка СИНТЕЗ, представляется в языке AMN конструкцией REFINEMENT:

```
REFINEMENT ObjectDM
```

Переменные, составляющие пространство состояний объектной модели, объявлены в разделе ABSTRACT_VARIABLES машины ObjectDM и типизируются в разделе INVARIANT:

```
ABSTRACT_VARIABLES
    typeNames, classNames, attributeNames,
    instanceType, typeAttributes,
      attributeType,
    unique, obligatory,
    intAttributeLowerBound,
      intAttributeHigherBound,
    objectIDs, objectType, objectsOfClass,
    integerAttributeValue,
    adtAttributeValue
INVARIANT ...
```

Раздел INVARIANT содержит формулу, которая состоит из предикатов, типизирующих переменные состояния и налагающих различные совместные ограничения на переменные. Предикаты соединяются операцией конъюнкции.

Так, имена типов и классов представлены переменными typeNames и classNames, тип которых — подмножество множества строк (STRING_Type):

```
typeNames: POW(STRING_Type) &
classNames: POW(STRING_Type)
```

Здесь POW — конструктор множества подмножеств.

Атрибуты (переменная attributeNames) представлены частичной функцией (знак «+–>»), ставящей в соответствие уникальному идентификатору атрибута (натуральному числу из множества NAT) имя атрибута (строку):

```
attributeNames: NAT +-> STRING_Type
```

Типы экземпляров классов (переменная instanceType) представлены тотальной функцией (знак –>) из множества имен классов в множество имен типов:

```
instanceType: classNames --> typeNames
```

Принадлежность атрибутов типам (переменная typeAttributes) выражена тотальной функцией из множества имен типов в множество подмножеств атрибутов:

```
typeAttributes:
  typeNames --> POW(dom(attributeNames))
```

Здесь dom — операция, возвращающая область определения функции, а dom(attributeNames) — множество имен атрибутов.

Типы значений атрибутов (переменная attributeType) представлены функцией из множества атрибутов в множество идентификаторов встроенных типов данных BuiltInTypes:

```
attributeType:
  dom(attributeNames) +-> BuiltInTypes
```

Множества уникальных атрибутов типов unique и множества определенных атрибутов типов obligatory представлены тотальными функциями из множества имен типов в множество подмножеств атрибутов:

```
unique:
  typeNames --> POW(dom(attributeNames))&
obligatory:
  typeNames --> POW(dom(attributeNames))
```

Нижние границы целочисленных атрибутов (переменная intAttributeLowerBound) представлены частичной функцией из множества атрибутов в множество целых чисел:

```
intAttributeLowerBound:
  dom(attributeNames) +-> INT
```

Аналогично представляются верхние границы.

Идентификаторы объектов (переменная objectIDs) представлены подмножеством натуральных чисел:

```
objectIDs: POW(NAT)
```

Типы объектов (переменная objectType) представлены тотальной функцией из множества объектных идентификаторов в множество имен типов:

```
objectType: objectIDs --> typeNames
```

Состав классов (переменная objectsOfClass) представлен тотальной функцией из множества имен классов в множество подмножеств идентификаторов объектов:

```
objectsOfClass:
  classNames --> POW(objectIDs)
```

Значения атрибутов объектов (переменные integerAttributeValue, adtAttributeValue и др.) представлены функциями из множества атрибутов в функции из множества идентификаторов объектов в множества значений атрибутов. Для простоты рассмотрены лишь функции для целочисленных атрибутов и атрибутов со значениями АТД (абстрактного типа данных) (объектами):

```
integerAttributeValue:
 dom(attributeNames) +-> (objectIDs+->INT)&
adtAttributeValue:
 dom(attributeNames) +-> (objectIDs+->NAT)
```

Дополнительные необходимые свойства переменных состояния представлены конъюнктивными компонентами инварианта. Так, каждый атрибут является атрибутом некоторого типа:

```
UNION(tt).(tt:typeNames|typeAttributes(tt))=
    dom(attributeNames)
```

Здесь UNION — родовая операция объединения, в данном случае объединяются множества атрибутов typeAttributes(tt) по всем именам типов tt из множества typeNames.

Никакой атрибут не принадлежит двум типам одновременно:

```
!(t1, t2).(t1: typeNames & t2: typeNames =>
  (typeAttributes(t1) /\ typeAttributes(t2)
    = {}))
```

Здесь «!» — знак квантора всеобщности, «=>» — логическая импликация, «/\» — символ пересечения множеств, «{}» — пустое множество.

Уникальные и определенные атрибуты типа выбираются из множества атрибутов типа:

```
!(tt).(tt: dom(unique) => unique(tt) <:
typeAttributes(tt)) &
!(tt).(tt: dom(obligatory) =>
    obligatory(tt) <: typeAttributes(tt))
```

Здесь «<:» — символ отношения множесво–подмножество.

Нижние и верхние границы могут быть определены только для целочисленных атрибутов:

```
!(attr).(attr: dom(intAttributeLowerBound)=>
    attributeType(attr) = Integer)
```

Тип объекта — экземпляра класса есть тип экземпляров этого класса:

```
!(cc).(cc: classNames =>
    !(oo).(oo: objectsOfClass(cc) =>
       objectType(oo) = instanceType(cc)))
```

Для каждого атрибута определена ровно одна функция значений:

```
dom(adtAttributeValue) /\
  dom(integerAttributeValue) = {} &
dom(adtAttributeValue) \/
  dom(integerAttributeValue) =
    dom(attributeNames)
```

Здесь «\/» — символ объединения множеств.

Для любого объекта и любого определенного атрибута типа этого объекта функция значений атрибута определена на объекте:

```
!(oo, aa).(oo: dom(objectType) &
  aa: typeAttributes(objectType(oo)) &
  aa: obligatory(objectType(oo)) =>
      (attributeType(aa) = Integer =>
       oo: dom(integerAttributeValue(aa))) &
      (attributeType(aa) = ADT =>
       oo: dom(adtAttributeValue(aa))))
```

Для любого объекта и любого целочисленного атрибута типа объекта, определенного на объекте и для которого определена нижняя (верхняя) граница, значение атрибута не меньше (не больше) нижней (верхней) границы:

```
!(oo, aa).(oo: objectIDs &
    aa: typeAttributes(objectType(oo)) &
    oo: dom(integerAttributeValue(aa) =>
    (aa: dom(intAttributeLowerBound) =>
        (integerAttributeValue(aa)(oo) >=
         intAttributeLowerBound(aa))) )
```

Объект однозначно идентифицируется набором своих уникальных атрибутов:

```
!(oo1, oo2).(oo1: objectIDs &
  oo2: objectIDs &
    objectType(oo1) = objectType(oo2) &
    unique(objectType(oo1)) /= {} &
    !(aa).(aa: unique(objectType(oo1)) =>
      (attributeType(aa) = Integer =>
        integerAttributeValue(aa)(oo1) =
         integerAttributeValue(aa)(oo2)) &
      (attributeType(aa) = ADT =>
         adtAttributeValue(aa)(oo1) =
          adtAttributeValue(aa)(oo2)) ) =>
    oo1 = oo2 )
```

Из всего ЯМД в спецификации рассмотрена единственная операция update обновления значений атрибута в объектах класса:

```
OPERATIONS
update(cls, attr, exp, cond) =
PRE cls: classNames &
  attr: typeAttributes(instanceType(cls)) &
  attributeType(attr) = Integer &
  exp: INT --> INT & cond: NAT --> BOOL
THEN
 integerAttributeValue :=
 integerAttributeValue <+
 { xx | xx: (NAT*(NAT<->INT)) &
  #(oo, val).( oo: objectsOfClass(cls) &
  val: INT &
    xx = attr |-> ({oo |-> val}) &
  (cond(integerAttributeValue(attr)(oo))
  = TRUE =>
  val=exp(integerAttributeValue(attr)(oo)))&
  (cond(integerAttributeValue(attr)(oo))
  = FALSE =>
  val=integerAttributeValue(attr)(oo)))}
END
```

Параметрами операции являются имя класса cls, имя целочисленного атрибута attr типа экземпляров класса, функция exp, отвечающая за преобразование атрибута, и функция cond, отвечающая условию на значение атрибута. Пусть o — некоторый объект класса cls, для которого определено значение атрибута attr, и это значение есть v. Тогда операция update изменяет значение атрибута на exp(v) в случае, если выражение cond(v) принимает значение «истина», и оставляет значение атрибута без изменений в противном случае. Очевидно, такая операция update есть обобщение примера обновления, рассмотренного в подразд. 2.2. Действительно, для рассмотренного примера cls есть source, attr есть val, exp(v) = - v, cond(v) = abs(v).

Заметим, что в рассмотренной спецификации для простоты не рассмотрены некоторые черты объектной модели, например отношения тип–подтип и класс–подкласс.

Спецификация, выражающая семантику модели ADM, представляется в языке AMN конструкцией

```
REFINEMENT ArrayDM
```

Переменные, составляющие пространство состояний объектной модели, объявлены в разделе ABSTRACT_VARIABLES машины ArrayDM:

```
ABSTRACT_VARIABLES
    arrayNames, dimensionNames,
    cellAttributeNames,
    arrayDimensions, arrayCellAttributes,
    cellAtrributeType, nullable,
    dimLowerBound, dimHigherBound,
    cells, dimensionValue,
    integerCellAttributeValue
```

Имена массивов представлены переменной arrayNames; имена измерений — переменной dimensionNames; имена атрибутов ячеек массива — переменной cellAttributeNames; принадлежность измерений массивам — переменной arrayDimensions; принадлежность атрибутов ячеек — переменной arrayCellAttributes; тип атрибута ячейки — переменной cellAtrributeType; атрибуты ячеек массивов, которые могут принимать неопределенные значения, — переменной nullable; верхние (нижние) границы измерений — переменной dimLowerBound (dimHigherBound); множества идентификаторов ячеек массивов — переменной cells, значения измерений в ячейках — переменной dimensionValue; значения атрибутов ячеек — переменной integerCellAttributeValue. Переменные типизируются в разделе INVARIANT при помощи частичных и тотальных функций аналогично переменным, использующимся для придания семантики объектной модели:

```
INVARIANT
   arrayNames: POW(STRING_Type) &
   dimensionNames: NAT +-> STRING_Type &
   cellAttributeNames: NAT +-> STRING_Type &
   arrayDimensions: arrayNames -->
   POW(dom(dimensionNames)) &
   arrayCellAttributes: arrayNames -->
     POW(dom(cellAttributeNames)) &
   cellAtrributeType:
     dom(cellAttributeNames) -->
       BuiltInTypes &
   nullable:
     dom(cellAttributeNames) --> BOOL &
   dimLowerBound:
     dom(dimensionNames) --> INT &
   dimHigherBound:
     dom(dimensionNames) +-> INT &
   cells: arrayNames --> POW(NAT) &
   dimensionValue:
     NAT*dom(dimensionNames) +-> INT  &
   integerCellAttributeValue:
     NAT*dom(cellAttributeNames) +-> INT &
```

Здесь «*» — знак декартова произведения множеств.

Дополнительные необходимые свойства переменных состояния представлены конъюнктивными компонентами инварианта. Так, любая ячейка любого массива однозначно идентифицируется набором значений измерений:

```
!(arr, cell1, cell2).(arr: arrayNames &
  cell1: cells(arr) &  cell2: cells(arr) &
  !(dim).(dim: arrayDimensions(arr) =>
    dimensionValue(cell1, dim) =
    dimensionValue(cell2, dim)) =>
    cell1 = cell2)
```

Для любой ячейки любого массива определены значения всех измерений и значение по крайней мере одного атрибута:

```
!(arr, cell).(arr: arrayNames &
 cell: cells(arr) =>
  !(dim).(dim: arrayDimensions(arr) =>
   (cell |-> dim): dom(dimensionValue)) &
   #(attr).(attr: arrayCellAttributes(arr) &
    cellAtrributeType(attr) = Integer &
    (cell, attr):
      dom(integerCellAttributeValue)) )
```

Аналогично объектной модели рассмотрена единственная операция ЯМД — операция обновления update:

```
OPERATIONS
update(cls, attr, exp, cond) =
PRE cls: arrayNames &
 attr: arrayCellAttributes(cls) &
  cellAtrributeType(attr) = Integer &
  exp: INT --> INT & cond: NAT --> BOOL
THEN
  integerCellAttributeValue :=
  integerCellAttributeValue <+
  { yy | yy: (NAT*NAT)*INT &
    #(cell, val).(cell: cells(cls) &
     val: INT &
    yy = ((cell |-> attr)|-> val) &
    (cond(integerCellAttributeValue(cell,
     attr)) = TRUE =>
      val =
       exp(integerCellAttributeValue(cell,
        attr))) &
      (cond(integerCellAttributeValue(cell,
       attr))= FALSE  =>
    val =
     integerCellAttributeValue(cell,attr)))}
END
END
```

Сигнатура операции совпадает с сигнатурой операции объектной модели. Семантика операции также аналогична: значение v атрибута attr массива cls заменяется на exp(v), если значение cond(v) есть «истина», и не изменяется в противном случае.

Заметим, что в данной спецификации для простоты не рассмотрены некоторые черты ADM, например нецелочисленные измерения.

Для формального доказательства того, что машина ArrayDM уточняет машину ObjectDM, необходимо построить инвариант уточнения, связывающий переменные машин, и добавить его к инварианту уточняющей машины.

Инвариант формализует принципы отображения ЯОД, изложенные в подразд. 2.1, и объединяет их в одну конъюнкцию.

Так, множество имен массивов совпадает с множеством имен классов:

```
classNames = arrayNames
```

Множество идентификаторов и имен измерений и атрибутов ячеек совпадает с множеством идентификаторов и имен атрибутов типов экземпляров классов:

```
attributeNames =
  dimensionNames \/ cellAttributeNames
```

Любому измерению любого массива соответствует атрибут типа экземпляра класса, соответствующего этому массиву:

```
!(arr, dim).(arr: arrayNames &
  dim: arrayDimensions(arr) =>
    #(attr).(attr:
     typeAttributes(instanceType(arr)) &
          attr = dim &
          attributeType(attr) = Integer) )s
```

Любому атрибуту ячейки любого массива соответствует атрибут типа экземпляра класса, соответствующего этому массиву, и типы атрибутов совпадают:

```
!(arr, cattr).(arr: arrayNames &
   cattr: arrayCellAttributes(arr) =>
    #(attr).(attr:
      typeAttributes(instanceType(arr)) &
         attr = cattr &
         attributeType(attr) =
           attributeType(cattr)))
```

Атрибут ячейки массива, который может принимать неопределенные значения, соответствует определенному (obligatory) атрибуту типа:

```
!(arr, cattr).(arr: arrayNames &
  cattr /: dom(nullable) &
    cattr: arrayCellAttributes(arr) =>
    cattr: obligatory(instanceType(arr)) )
```

Здесь знак «/:» обозначает отношение непринадлежности элемента множеству.

Измерения соответствуют уникальным атрибутам типов:

```
!(arr, dim).(arr: arrayNames &
    dim: arrayDimensions(arr) =>
      dim: unique(instanceType(arr)) )
```

Верхние (нижние) границы измерений равны верхним (нижним) границам соответствующих атрибутов типов:

```
!(dim).(dim: dom(dimLowerBound) =>
    dim: dom(intAttributeLowerBound) &
    dimLowerBound(dim) =
      intAttributeLowerBound(dim))
```

Непустые ячейки массивов соответствуют объектам классов:

```
cells = objectsOfClass
```

Для любой ячейки значения ее измерений и определенных атрибутов совпадают со значениями соответствующих атрибутов объекта, соответствующего ячейке:

```
!(cell, dim).(cell: NAT & dim: NAT &
  (cell |-> dim): dom(dimensionValue) =>
  cell: dom(integerAttributeValue(dim)) &
  dimensionValue(cell, dim) =
    integerAttributeValue(dim)(cell)) &
!(cell, cattr).(cell: NAT & cattr: NAT &
   (cell |-> cattr):
   dom(integerCellAttributeValue) =>
   cell: dom(integerAttributeValue(cattr)) &
   integerCellAttributeValue(cell, cattr) =
     integerAttributeValue(cattr)(cell) )
```

Для указания того, что машина ArrayDM уточняет машину ObjectDM, в машину ArrayDM была добавлена директива

```
REFINES ObjectDM
```

Спецификации ObjectDM и ArrayDM вместе с инвариантом уточнения были загружены в инструментальное средство Atelier B [23]. Автоматически были сгенерированы теоремы, выражающие уточнение спецификаций. В частности, для операции update были сгенерированы 10 теорем. Три из них были доказаны автоматически, для доказательства остальных необходимо применять интерактивные средства доказательства.

# 4 Родственные исследования и направления дальнейшей работы

Родственными данной работе следует считать исследования, связанные с отображением моделей,

основанных на многомерных массивах, в реляционную модель данных. Обычно они нацелены на реализацию многомерных массивов при помощи реляционных СУБД. Такие работы начались одновременно с исследованиями моделей, основанных на многомерных массивах [6], и продолжаются в настоящее время [24].

Основные особенности данной работы состоят в следующем. В качестве исходной модели при отображении используется специфическая модель, основанная на многомерных массивах СУБД SciDB, язык которой представляет собой комбинацию декларативного SQL-подобного языка и функционального языка, включающего специфические операции над многомерными массивами. В качестве целевой модели используется объектная модель с Datalog-подобным языком запросов (программ) — язык СИНТЕЗ. Для отображения обеспечивается формальное доказательство сохранения информации и семантики операций ЯМД.

Отметим, что результаты работы могут быть с легкостью обобщены и использованы при интеграции в системах, использующих каноническую модель, отличную от языка СИНТЕЗ, например другую объектную (ODMG) или объектно-реляционную модель (SQL:2003). Результаты также могут быть использованы для интеграции ресурсов, представленных в модели, основанной на многомерных массивах, но отличной от ADM.

Некоторые вопросы отображения требуют дальнейших исследований. Например, следует ли иметь в канонической модели при интеграции масссив-ориентированных моделей данных операции, связанные с размером порции (chunk size) данных в БД [10]?

Дальнейшую работу можно разбить на два этапа:

(1) расширение инструментальных средств поддержки предметных посредников для виртуальной интеграции SciDB-ресурсов:

   (а) расширение средств регистрации ресурсов в посреднике [11] трансформацией ЯОД ADM в каноническую модель;

   (б) создание SciDB-адаптера — специального ПО, связывающего исполнительную среду посредников с SciDB-ресурсами (составной частью адаптера является разработанная трансформация ЯМД);

(2) применение технологии предметных посредников для решения научных задач в некоторой предметной области над множеством неоднородных ресурсов, включающим SciDB-ресурсы.

Автор выражает благодарность Л. А. Калиниченко, П. Е. Велихову и А. Е. Вовченко за полезные замечания, высказанные в ходе обсуждения данной работы на семинарах ИПИ РАН.

## Литература

1. Challenges and opportunities with big data // A community white paper developed by leading researchers across the United States, 2012. http://cra.org/ccc/docs/init/bigdatawhitepaper.pdf.
2. *Abrial J.-R.* The B-Book: Assigning programs to meanings. — Cambridge: Cambridge University Press, 1996.
3. *Vassiliadis P., Sellis T. K.* A survey of logical models for OLAP databases // SIGMOD Record, 1999. Vol. 28. No. 4. P. 64–69.
4. *Pedersen T. B., Jensen C. S.* Multidimensional database technology // IEEE Computer, 2001. Vol. 34. No. 12. P. 40–46.
5. *Libkin L., Machlin R., Wong L.* A query language for multidimensional arrays: Design, implementation, and optimization techniques. — SIGMOD, 1996. P. 228–239.
6. *Baumann P.* A database array algebra for spatio-temporal data and beyond // Next generation information technologies and systems. Lectures notes in computer science ser. Springer Verlag KG, 1999. Vol. 1649. P. 76–93.
7. Overview of SciDB: Large scale array storage, processing and analysis. The SciDB development team. — SIGMOD, 2010.
8. Large synoptic survey telescope. http://www.lsst.org.
9. *Becla J., Lim K.-T.* Report from the First Workshop on Extremely Large Databases // Data Sci. J., 2008. Vol. 7.
10. SciDB User's Guide. Version 12.3, 2012. http://www.scidb.org.
11. *Kalinichenko L. A., Briukhov D. O., Martynov D. O., Skvortsov N. A., Stupnikov S. A.* Mediation framework for enterprise information system infrastructures // Volume Databases and Information Systems Integration: 9th Conference (International) on Enterprise Information Systems (ICEIS 2007) Proceedings — Funchal, 2007. P. 246–251.
12. *Захаров В. Н., Калиниченко Л. А., Соколов И. А., Ступников С. А.* Конструирование канонических информационных моделей для интегрированных информационных систем // Информатика и её применения, 2007. Т. 1. Вып. 2. С. 15–38.
13. *Kalinichenko L. A., Stupnikov S. A.* Heterogeneous information model unification as a prerequisite to resource schema mapping // Information Systems: People, Organizations, Institutions, and Technologies: 5th Conference of the Italian Chapter of Association for Information Systems itAIS Proceedings. — Berlin–Heidelberg: Springer Physica Verlag, 2010. P. 373–380.
14. *Kalinichenko L. A., Stupnikov S. A., Martynov D. O.* SYNTHESIS: A language for canonical information modeling and mediator definition for problem solving in heterogeneous information resource environments. — Moscow: IPI RAN, 2007. 171 p.

15. *Брюхов Д. О., Вовченко А. Е., Захаров В. Н., Желенкова О. П., Калиниченко Л. А., Мартынов Д. О., Скворцов Н. А., Ступников С. А.* Архитектура промежуточного слоя предметных посредников для решения задач над множеством интегрируемых неоднородных распределенных информационных ресурсов в гибридной грид-инфраструктуре виртуальных обсерваторий // Информатика и её применения, 2008. Т. 2. Вып. 1. С. 2–34.
16. *Kersten M. L., Zhang Y., Ivanova M., Nes N.* SciQL, a query language for science applications // EDBT/ICDT — Workshop on Array Databases 2011 Proceedings. — Uppsala, Sweden, 2011. P. 1–12.
17. *Abrial J.-R.* The B-Book: Assigning programs to meanings. — Cambridge: Cambridge University Press, 1996.
18. Astronomy in ArrayDB. http://trac.scidb.org/raw-attachment/wiki/UseCases/Astronomy%20in%20ArrayDB.pdf
19. ATL Project. http://www.eclipse.org/m2m/atl.
20. *Budinsky F., Steinberg D., Ellersick R., Grose T.* Eclipse modeling framework. Ch. 5: Ecore modeling concepts. — Addison Wesley Professional, 2004.
21. Meta Object Facility (MOF) 2.0 Core Specification, 2003. http://www.omg.org/cgi-bin/apps/doc?ptc/03-10-04.pdf.
22. *Kalinichenko L. A.* Method for data models integration in the common paradigm // 1st East-European Symposium on Advances in Databases and Information Systems ADBIS'97 Proceedings. — St.-Petersburg: Nevsky Dialect, 1997. Vol. 1: Regular papers. P. 275–284.
23. Atelier B: The industrial tool to efficiently deploy the B Method. http://www.atelierb.eu/index-en.php.
24. *Van Ballegooij A.* RAM: Array database management through relational mapping // SIKS Dissertation ser. No. 2009-25. http://oai.cwi.nl/oai/asset/14074/14074D.pdf.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 35–39

# ИССЛЕДОВАНИЕ ГРАФА КАТЕГОРИЙ АНГЛИЙСКОЙ ВЕРСИИ ВИКИПЕДИИ*

А. В. Шкотин[1]

**Аннотация:** Википедия является выдающимся проектом по накоплению знаний как общего пользования, так и различных областей специализации. Проверка качества этих знаний, особенно автоматическая, чрезвычайно важна. В работе представлены результаты изучения строения английской версии ГКВ (орграфа категориальных статей Википедии). Являясь по своей идее системой тем, он поддерживает систематизацию знаний, и представляет интерес, из чего эта систематизация состоит и как она устроена. Показано, что в графе есть неприемлемые логические нарушения, и обсуждаются организационные и технические методы их устранения.

**Ключевые слова:** Википедия; орграф; связные компоненты; логический анализ

## 1 Введение

Орграф категориальных статей Википедии [1] есть подграф графа, в котором статьи Википедии приписаны категориальным статьям. Выделение ГКВ из этого полного графа есть первая техническая задача. Важно, что далее изучается строение ГКВ на некоторый момент времени и в нем есть незавершенная, «строящаяся» часть. Поэтому выводы надо делать с осторожностью. Естественно ввести термин «точка роста», когда натыкаешься в ГКВ на часть, которая еще не завершена. Дамп полного графа получен из ИСП РАН в виде двух текстовых файлов: файла отображения номера страницы Википедии в номер категориальной страницы, что приписывает страницу категории, и файла, в котором номеру страницы Википедии приписано ее наименование. Математически ГКВ есть орграф, каждый узел которого взаимно однозначно соответствует категориальной странице и помечен ее номером. Стрелка (дуга) из узла $N_1$ в узел $N_2$ идет тогда и только тогда, когда страница с номером $N_1$ есть подкатегория страницы с номером $N_2$. Всего таких стрелок 1 221 133.

Множество узлов ГКВ (593 796), как и любого произвольного графа, распадается на два подмножества: изолированные узлы (26 272) и узлы, связанные стрелками (567 24). Изолированная категория — это, скорее всего, «точка роста»: на момент снятия дампа она уже есть, но стрелок еще нет.

Далее анализируется только «граф стрелок», т. е. все характеристики даны без учета изолированных узлов. Состав изолированных узлов можно посмотреть в отчете [2] (далее — отчет) в таблице, указанной во введении. Состав и характеристики узлов со стрелками можно посмотреть в таблице, указанной там же, равно как и граф стрелок. Важный вопрос — количество связных компонент графа, так как в дальнейшем их строение можно изучать отдельно. Таких компонент оказалось 1987. Изолированные узлы при этом учитываются отдельно. Алгоритм разбиения описан в отчете [3]. Впрочем, проще воспользоваться пакетом программ, например Pajek [4], который умеет разбивать узлы графа на слабо связные компоненты.

Первые 10 самых больших компонент указаны в табл. 1, где $C_n$ — уникальный номер компоненты, присвоенный при разбиении. Конечно, в случае с Википедией малые компоненты — это точки роста. Петель ($N_1 \to N_1$) в графе нет.

**Таблица 1** Объем связных компонент

| $C_n$ | Количество |
|---|---|
| 1 | 561 636 |
| 21 727 | 210 |
| 14 332 | 36 |
| 2 863 | 29 |
| 20 842 | 27 |
| 6 680 | 20 |
| 19 212 | 19 |
| 20 868 | 19 |
| 13 325 | 17 |
| 13 287 | 16 |

*Статья рекомендована к публикации в журнале Программным комитетом конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» (RCDL-2012).

[1]Государственный геологический музей им. В. И. Вернадского Российской академии наук, отдел ГИС, ashkotin@acm.org

Источников (узлов, в которые нет входящих стрелок) — 345 597. Это категории нижнего уровня. Стоков (узлов, из которых нет исходящих стрелок) — 11 767. Это категории верхнего уровня дампа и, скорее всего, точки роста. Промежуточных узлов, соответственно, 210 160.

Максимальное количество исходящих из одного узла стрелок — 85. Речь идет о промежуточном узле № 690451 с заголовком Category:World War II, т. е. эта категория приписана 85 надкатегориям. Максимальное количество входящих стрелок (12 625) имеет промежуточный узел № 692309 с проясняющим заголовком Category:Albums by artist.

## 2 Анализ заголовков

Заголовки всех узлов категорий (включая изолированные) можно посмотреть в отчете в таблице, указанной в разделе «Анализ заголовков». Таблица содержит 584 606 узлов. Следовательно, 9190 узлов ГКВ не имеют заголовков. Они ждут своего исследователя. Анализ текстов заголовков, даже безотносительно к их подчинению, — отдельная увлекательная задача. Но начать надо с использованного состава букв.

### 2.1 Алфавит

Рассмотрим состав букв (characters), употребленных при именовании категориальных статей. Текстовый файл (UTF-8), содержащий состав алфавита можно посмотреть в прикреплении cat2title.abc0.txt к отчету. Как разделитель букв используется знак «|». Этот алфавит представлен на рис. 1. В приложении id2title.abc.txt к отчету можно найти впечатляющее разнообразие букв заголовков всех статей Википедии.

### 2.2 Термины в заголовках

Это может стать отдельным важным исследованием. Например, количество заголовков, в которых встречается слово album, — 17 591.

## 3 Стоки

В приложении 1 к отчету можно посмотреть начало таблицы стоков с самым большим количеством входящих стрелок.

В приложении 2 к отчету можно посмотреть путь-рекордсмен, предоставленный Антоном Коршуновым из ИСП РАН.

Самый длинный путь — 294 вершины. Его начальная категория — № 5760285 Category:Anastacia songs, а конечная — № 691484 Category:Music.

## 4 Строение орграфа категориальных статей Википедии в целом

В работе [5, с. 9] указывается, что в ГКВ есть циклы. По идее, циклы — это аномалии на графе подчинения категорий, они должны занимать малую его часть. Назовем для краткости объединение орциклов графа и орпутей между циклами — *ядром*, а дополнительную часть графа — *мантией*. Стрелки же между мантией и ядром назовем *связующими*. Таким образом, в целом граф состоит из ядра, мантии и связующих стрелок, часть из которых идет из ядра в мантию, а часть — из мантии в ядро. Чтобы выделить ядро, был применен следующий алгоритм:

(1) находим в графе источники и стоки и удаляем их;

(2) если в графе не осталось узлов, то стоп (ядра нет);

(3) если есть источники или стоки, то идти на (1). Если нет, то стоп (в графе осталось только ядро).

## 5 Ядро орграфа категориальных статей Википедии

В строении ядра важно, что пути между циклами, сами не входящие в циклы, составляют самостоятельную интересную часть ядра.

```
| |!|"|&|'|(|)|*|+|,|-
|./|0|1|2|3|4|5|6|7|8|9|:|;|?|@|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P
|Q|R|S|T|U|V|W|X|Y|Z|a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v
|w|x|y|z|~|¡|ª|°|µ|·|º|½|Á|Â|Ä|Å|Æ|Ç|É|Í|Î|Ñ|Ó|Õ|Ö|×|Ø|Ú|Ü|Þ|ß|à
|á|â|ã|ä|å|æ|ç|è|é|ê|ë|ì|í|î|ï|ð|ñ|ò|ó|ô|õ|ö|ø|ù|ú|û|ü|ý|þ|ÿ|Ā|ā
|ă|ą|Ć|ć|Č|č|ď|Đ|đ|ē|ė|ę|ě|ğ|ħ|ħ|Ī|ī|İ|ı|ļ|Ľ|ľ|Ł|ł|ń|ņ|ň|Ō|ō|ő|œ
|ř|Ś|ś|Ş|ş|Š|š|Ţ|ţ|ť|ū|ŭ|ů|ŵ|ź|Ż|ż|Ž|ž|ơ|Ş|‘|,|ļ|ķ|ţ|ạ|ả|ậ|ắ|ế|ề|ể|ễ
|ệ|ị|ọ|ộ|ụ|ỳ|ỹ|-|-|‘|’|…|共|台|和|國|歲|灣|萬|!
```

**Рис. 1** Состав используемых символов

## 5.1 Состав ядра

Количество стрелок в ядре — 38 538. Узлов же — 13 545. Граф ядра опубликован в таблице, указанной в разделе «Состав ядра» отчета. Далее было традиционно выполнено «расщепление» ядра на связные компоненты. Оказалось, что имеются одна большая компонента — 13 507 узлов и еще 19 пар узлов. Характеристики узлов ядра, включая разбиение на связные компоненты, можно посмотреть в таблице, указанной в разделе «Состав ядра» отчета.

Рассмотрим компоненту № 764 ядра. Это пример пары, которая является даже связной компонентой не только ядра, а самого ГКВ. В компоненте два узла:

(1) № 28736601, Category:Wikipedia sockpuppets of ShantanuSingh198 ;

(2) № 28736686, Category:Suspected Wikipedia sockpuppets of ShantanuSingh19.

В Википедии они также ссылаются только друг на друга.

**Анализ.** Что бы ни обозначало «Wikipedia sockpuppets of ShantanuSingh198», очевидно, что нечто, под него подпадающее (как под понятие), не может быть одновременно лишь «подозреваемым» на подпадание. Равно как и наоборот, т. е. логически эти две категории не пересекаются и обе стрелки должны быть удалены. Отношения же между ними, например, на OWL 2 [6] должно было бы быть таким:

```
DisjointClasses(wcg:Wikipedia_sockpuppets_of_
ShantanuSingh198,
wcg:Suspected_Wikipedia_sockpuppets_of_
ShantanuSingh198)
```

При этом правильнее ссылаться в обеих статьях друг на друга через тег Википедии «See also».

## 5.2 Сильно связные компоненты ядра

В ядре представляет интерес зацикливание отношения подкатегория–надкатегория. Тут есть два подхода:

(1) общий — применить алгоритм поиска сильно связных компонент (ССК);

(2) частный — найти так называемые «линзы» — два узла, ссылающихся друг на друга (как подкатегория–надкатегория).

Второй путь вполне приемлем для ГКВ, так как, по идее, в нем вообще не должно быть циклов. Впрочем, как в отношении линз, так и в отношении циклов большей длины следует заметить, что они математически утверждают эквивалентность соответствующих терминов, т. е. синонимию, что в принципе возможно. Но конкретно в Википедии возможна реализация через redirect. Интуитивно же в большинстве случаев обнаруживается ошибка, т. е. какие-то стрелки цикла ошибочны.

Чтобы получить состав сильно связных компонент ядра, была использована программа Pajek [4]. Заметим, что петель в ГКВ нет, а поэтому узлы ядра, не попавшие в ССК, — это узлы на путях между циклами (см. выше).

Сильно связных компонент оказалось 457. Узлов, не входящих в ССК, так сказать, связующих ядра, — 7646. Есть одна гигантская по сравнению с остальными ССК — в ней 3967 узлов.

В отчете в разделе «Сильно связные компоненты ядра» приведена таблица самых больших ССК. Рассмотрим для примера компоненту № 41, у которой всего 9 узлов (табл. 2, рис. 2).

Если номер накладывается на стрелку, то под ним наконечника (треугольничка) нет. Это важно, так как Pajek рисует «линзы» ($У_1 \to У_2 \to У_1$) как одну стрелку с наконечниками на обоих окончани-

**Таблица 2** Заголовки узлов ССК № 41

| Номер узла | Заголовок |
|---|---|
| 717 227 | Category:Orthodox rabbis |
| 717 302 | Category:Talmud rabbis |
| 799 461 | Category:Mishnah |
| 799 587 | Category:Talmud |
| 6 110 893 | Category:Talmudists |
| 8 398 752 | Category:Talmud people |
| 11 334 178 | Category:Rabbinic literature |
| 15 249 105 | Category:Talmud concepts and terminology |
| 26 795 615 | Category:Chazal |

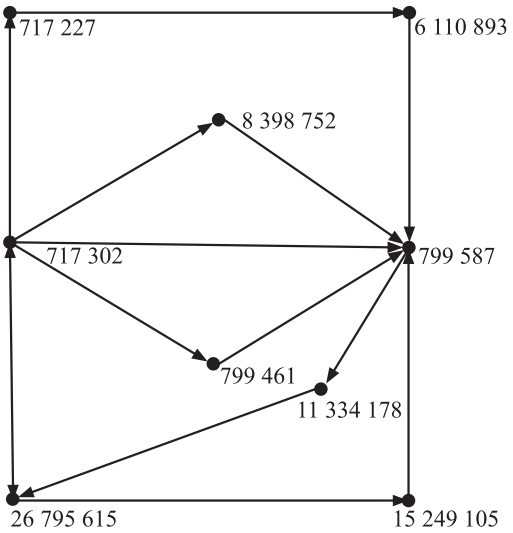



**Рис. 2** Рисунок графа ССК № 41

ях. В данной ССК (см. рис. 2) линза всего одна — слева внизу вертикально.

### 5.3 Линзы

Линза — это два узла таких, что $У_1 \to У_2$ и $У_2 \to \to У_1$. Она может быть отдельной ССК, а может входить в ССК как часть.

В ядре оказалось 1269 линз. Из них 1260 имеют заголовки для обоих узлов. Их можно посмотреть в таблице, указанной в разделе «Линзы» отчета.

## 6 Мантия — ациклическая часть орграфа категориальных статей Википедии

Чтобы получить мантию, удалим из ГКВ ядро. При этом оказывается, что часть источников и стоков стали изолированными. В первом случае все исходящие из них стрелки попали в ядро, во втором — все входящие в них стрелки шли из ядра. Изолировавшихся источников — 14 421, стоков — 60.

Кроме того, в мантии появляются ложные вершины (пики). Это те ее узлы, которые стали стоками после удаления ядра, а вообще-то имели исходящие стрелки, которые все попадали в ядро. Таких вершин 18 157. Причем максимальная высота — 28. Для сравнения, стоков ГКВ, получивших уровень, т. е. неизолированных — 11 707, максимальная высота — 24.

Ложная вершина-рекордсмен (высоты 28) имеет № 15 715 670, а заголовок — Category:Creation myths.

**Замечание.** Конечно, ГКВ можно представить и в виде «галстука-бабочки», как в работе [7], где орграф был использован для представления схемы связей между транснациональными корпорациями. Но в данном случае нагляднее сравнение с горами — вверх к более обширным темам, горами, в которых есть ядро из 20 связных компонент, одна из которых большая, а 19 — линзы.

Число узлов на уровнях показано в табл. 3 и оправдывает сравнение с горами. В строке NULL указано количество изолированных узлов мантии, а в строке 0 — количество узлов в ядре.

### Связующие стрелки

Между мантией и ядром есть стрелки — связующие. Стрелок из ядра в мантию — 591. Стрелок из мантии в ядро — 210 514.

**Таблица 3** Распределение узлов по уровням

| Уровень | Количество узлов |
|---|---|
| NULL | 14 481 |
| 28 | 1 |
| 27 | 2 |
| 26 | 3 |
| 25 | 3 |
| 24 | 5 |
| 23 | 7 |
| 22 | 12 |
| 21 | 16 |
| 20 | 20 |
| 19 | 30 |
| 18 | 50 |
| 17 | 57 |
| 16 | 71 |
| 15 | 100 |
| 14 | 149 |
| 13 | 226 |
| 12 | 425 |
| 11 | 697 |
| 10 | 1 187 |
| 9 | 1 915 |
| 8 | 3 103 |
| 7 | 4 858 |
| 6 | 7 754 |
| 5 | 13 019 |
| 4 | 23 302 |
| 3 | 45 323 |
| 2 | 105 958 |
| 1 | 331 205 |
| 0 | 13 545 |

## 7 Обсуждение

1. Влияние ядра на строение мантии оказывается существенным. Так, ложный сток имеет высоту 28 при максимальной высоте настоящего стока 24. Такое может случиться, только если ядро находится на «вершине» мантии. Кроме того, появилось 29 ложных источников — в них шли стрелки только из ядра. Количество ложных стоков — 18 157. При этом ядро состоит всего лишь из одной большой связной компоненты и 19 линз.
2. Хотя «физически» отдельное изучении мантии оправдано, для совокупного строения графа лучше не удалять ядро из ГКВ, а свернуть его ССК в «тяжелые» узлы, пометив их количеством узлов в ССК. Такой тяжелый узел наследует все внешние ССК стрелки, а образовавшиеся петли внутренних стрелок стоит удалить. Тогда получится ациклический граф, так как ССК не могут образовывать цикл. Уровни такого графа с указанием распределения по ним тяжелых узлов дадут

более реалистичную картину ГКВ. При этом есть все основания полагать, что по крайней мере одна из тяжелых вершин будет стоком.

3. В растущем графе, которым по преимуществу является ГКВ, «точки роста» (изолированные узлы; все связные компоненты, кроме главной; стоки вне главной связной компоненты) не представляют особого интереса. Поэтому стоит сразу выделить главную связную компоненту и изучать только ее.

## 8 Другие способы исследования

Можно напрямую изучать http://dbpedia.org через точку входа для SPARQL: http://dbpedia.org/sparql. Привязка к категории идет через свойство http://purl.org/dc/terms/subject.

Вот пример запроса, который начинает выдавать полный граф связи страниц и категорий:

```
select ?x ?z where {?x dcterms:subject ?z}
```

Надо только поставить timeout, например, 1000.

Запрос, выдающий отношение «x is a sub-category of z» (см. с. 5 «Categories» [5]):

```
select ?x ?z where {?x skos:broader ?z}
```

А вот запрос, выдающий «линзы»:

```
select ?x ?z where {?x skos:broader ?z. ?z
skos:broader ?x.}
```

Вот узлы первой:

http://dbpedia.org/resource/Category:Political_philosophers;
http://dbpedia.org/resource/Category:Political_theorists.

Она действительно есть в Википедии.

А всего запрос выдает 2000 линз, что, наверное, не предел.

## 9 Заключение

Естественно считать, что ГКВ должен быть ациклическим графом. Таким образом, исследование показало, что аномалии значительны.

Можно создать средства, которые, обнаруживая аномалию, например линзу, будут размещать на соответствующих страницах в Discussion уведомление о возможном логическом противоречии.

Основных вопросов два.

1. Как к такому подходу отнесутся авторы страниц категорий? Это можно проверить экспериментально.
2. Как к логическим противоречиям относятся идеологи Википедии? Те, кто задает правила классификации? Судя по всему, индифферентно.

Общая рекомендация: многие отношения между категориями, попавшие в «sub-category of», следует перенести в «See also».

Оценить предстоящую работу можно так: для начала надо разобраться с 1269 линзами. Они значительно убавят размер ССК.

Только если это нужно википедистам, можно было бы продолжить работу в следующих направлениях:

- исследовать длинные пути;
- попытаться представить архитектуру графа в целом (например, применить трехмерную визуализацию);
- проанализировать состав и логику связи заголовков (особенно ССК).

Особняком стоит задача получить и проанализировать русский ГКВ. В проекте DBpedia можно получить дамп русской версии, надо только перекодировать буквы с rdf-кодов (например, \u0432) в UTF-8.

## Литература


1. *Korshunov A., Turdakov D., Jeong J., Lee M., Moon Ch.* A category-driven approach to deriving domain specific subset of Wikipedia // SYRCoDIS'11: 7th Spring Researchers Colloquium on Databases and Information Systems Proceedings, 2011. P. 43–53.
2. *Шкотин А.* Исследование графа категорий английской версии Wikipedia. Сообщение о результатах первого этапа. 2011. http://sites.google.com/site/alex0shkotin/grafy/wikipedia-category-graph.
3. *Шкотин А.* Разбиение графа на связные компоненты: Алгоритм и программа. 2011. http://sites.google.com/site/alex0shkotin/grafy/svaznye-komponenty.
4. *Batagelj V., Mrvar A.* Pajek: Program for analysis and visualization of large networks: Reference manual. — Ljubljana: University, 2012.
5. *Bizer C., Lehmann J., Kobilarov G., Auer S., Becker C., Cyganiak R., Hellmann S.* DBpedia — a crystallization point for the Web of Data // J. Web Semantics, 2009. Vol. 7. No. 3. P. 154–165.
6. OWL 2 Web Ontology Language: Structural specification and functional-style syntax: W3C recommendation / Eds. B. Motik, P. F. Patel-Schneider, B. Parsia. 2009. http://www.w3.org/TR/owl2-syntax.
7. *Vitali S., Glattfelder J. B., Battiston S.* The network of global corporate control // Cornell University Library (submitted on July 28, 2011 (v1), last revised Sep. 19, 2011 (this version, v2)). http://arxiv.org/abs/1107.5728.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 40–55

# МЕТОДЫ АКТИВНОЙ АУТЕНТИФИКАЦИИ НА ОСНОВЕ АНАЛИЗА ДИНАМИКИ РАБОТЫ ПОЛЬЗОВАТЕЛЕЙ С КЛАВИАТУРОЙ

В. Ю. Каганов[1], А. К. Королёв[2], М. Н. Крылов[3], И. В. Машечкин[4], М. И. Петровский[5]

**Аннотация:** Проведен обзор некоторых эффективных методов аутентификации на основе поведенческих моделей пользователей, построенных с использованием данных, полученных при анализе работы пользователя с клавиатурой. Также предложен новый подход к представлению собираемых данных, проведены эксперименты с использованием этого представления и различных алгоритмов машинного обучения.

**Ключевые слова:** аутентификация; машинное обучение; деревья решений; клавиатура; потенциальные функции; поведенческий анализ

## 1 Введение

В большинстве современных информационных систем одной из важнейших задач, помимо сохранения и обработки данных, является задача разграничения доступа к ресурсам. Это необходимо как для предотвращения несанкционированного доступа к системам извне, так и для разграничения прав сотрудников, работающих с информационной системой внутри организации. Поэтому задача аутентификации, т. е. проверки подлинности пользователя, желающего получить доступ к системе, является одной из ключевых.

В настоящее время эта задача может быть решена множеством различных способов. Пожалуй, самый популярный способ из тех, что используются в современных информационных системах, — это аутентификация по паролю — специальной последовательности символов, не известной никому, кроме пользователя, которому разрешен доступ к системе. Помимо очевидных преимуществ, таких как простота реализации и использования, а также распространенность, у нее есть и существенные недостатки: сама кодовая фраза может быть забыта, передана другим лицам, а при недостаточной длине или сложности — подобрана простым перебором или перебором по словарю, что ставит под сомнение возможность их использования в системах, требующих высшего уровня безопасности.

От последнего недостатка свободны системы электронно-цифровых подписей (ЭЦП), также применяемых для аутентификации. Однако проблему безопасного хранения закрытых ключей таким образом решить невозможно.

В связи с вышесказанным значительная часть систем, обеспечивающих эффективную безопасность, использует биометрию для определения личности пользователя. Биометрические комплексы могут быть разделены на две категории:

(1) системы, которые используют различные в силу естественных причин особенности человека, такие как отпечатки пальцев, сетчатка глаза, голос, тепловая карта тела и т. п. Они эффективны, но довольно дороги, так как требуют установки специального оборудования;

(2) системы, которые анализируют поведение пользователя и основаны на опыте или особых навыках. Они не требуют какого-либо специального оборудования и просты для внедрения. Один из таких подходов – анализ динамики нажатий клавиш.

В этой статье будут рассмотрены подходы к аутентификации пользователя по различным моделям, построенным на информации о нажатиях клавиш.

[1]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, vladhid@mlab.cs.msu.su

[2]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, akorolev@mlab.cs.msu.su

[3]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, krylovm@mlab.cs.msu.su

[4]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, mash@cs.msu.su

[5]Московский государственный университет им. М. В. Ломоносова, факультет вычислительной математики и кибернетики, michael@cs.msu.su

# 2 Постановка задачи

## 2.1 Способы аутентификации

В рамках проблемы рассматриваются следующие методы анализа динамики нажатий клавиш пользователем:

– **статическая аутентификация при входе.** Анализ основывается на известном шаблоне, слове или другом заранее предопределенном тексте. Набираемые пользователем при входе данные собираются (например, при вводе пароля) и сравниваются с предшествующими удачными попытками входа. Данный подход рассматривается как расширение стандартного метода аутентификации при входе с использованием логина/пароля (т. е. при входе в систему проверяется не только *что* набрал пользователь, но и *как* он это сделал). Стоит отметить следующие особенности статической аутентификации:

  • **небольшое количество входных данных.** Как правило, статическая аутентификация работает в паре с аутентификацией по паролю, а использование чрезвычайно длинных паролей, которые пользователь набирал бы вручную (более 100 символов), практически исключено;

  • **однообразие данных.** Один и тот же пароль, как правило, используется для входа в систему множество раз, и из набора этого пароля можно извлечь лишь небольшое количество признаков. Таким образом, метод должен быть оптимизирован для распознавания пользователя по небольшому множеству параметров;

  • **высокая скорость работы.** В случае статической аутентификации нет возможности проводить обработку данных для аутентификации параллельно с работой пользователя. До тех пор пока аутентификация не завершится успешно, пользователь не будет допущен к работе с системой. Поэтому необходимо сделать задержку между вводом пароля и входом в систему как можно меньшей.

– **периодическая динамическая аутентификация.** В динамических методах происходит аутентификация пользователя по его работе с клавиатурой *во время* сессии работы с системой. В данном случае процесс проверки пользователя может запускаться по какому-либо событию, например по времени или при обнаружении потенциально подозрительной активности. Собранные в рамках сессии данные сравниваются с поведением пользователя в предшествующих сессиях для выявления аномалий. Данный метод имеет ряд преимуществ перед статическим анализом. Во-первых, набираемый текст может быть произвольным – анализируется только то, *как* пользователь его набирает. Во-вторых, такие данные значительно проще собрать: в общем случае они собираются в фоновом режиме при обычной работе пользователя. Таким образом обучить систему значительно проще и, кроме того, на большом массиве данных даже у неопытного пользователя проще выделить характерные признаки;

– **непрерывная динамическая аутентификация.** Является дополнением периодической аутентификации в том, что проверка подлинности запускается в фоновом режиме и выполняется постоянно (с точностью до минимальной порции данных, на которой возможна корректная работа алгоритма классификации).

## 2.2 Формальная постановка задачи

Для всех перечисленных способов формальная постановка задачи звучит одинаково.

Пусть задано некоторое множество *пользователей* $\mathfrak{U} = \{U_1, U_2, \ldots, U_n\}$. Также введем понятие *действия* пользователя $A$, которым может считаться, например, нажатие одной клавиши или набор пароля. Задача обучения в этом случае заключается в том, чтобы каждому пользователю $U_i \in \mathfrak{U}$ сопоставить некоторую функцию (модель) $F_i$, которая может служить мерой аномальности действия $A_i^{\text{next}}$ для пользователя $U_i$. В таком случае задача аутентификации — это вычисление $F_i(A_i^{\text{next}})$ и обработка полученного значения, по которому принимается решение об успешности аутентификации пользователя $U_i$.

# 3 Обзор существующих методов статической аутентификации

Интересным представляется множество подходов, описанных в статье [1]. Из них стоит выделить 4 эксперимента, проведенных авторами статьи, которые были направлены на оценку качества тех или иных моделей представления данных.

## 3.1 Модель, основанная на измерении времени удержания

Одним из наиболее существенных признаков, извлекаемых из данных о динамике работы поль-

зователя с клавиатурой, является время удержания пользователем различных кнопок в нажатом состоянии. Априори предполагается, что у разных пользователей эти времена будут заметно различаться, и по величине отклонения этого времени можно будет установить аномальность действия $A_i^{\text{next}}$.

В этом методе в качестве модели берется вектор $X$ из $n$ элементов, каждый из которых соответствует одной кнопке на клавиатуре и является парой $(M_k, D_k)$: элемент $M_k$ — среднее время удержания клавиши $k$, а $D_k$ — величина стандартного отклонения для клавиши $k$.

Таким образом, действие пользователя $A_i^{\text{next}}$ — время удержания некоторой клавиши $K$ – признается аномальным, если оно отличается от среднего $M_K$ на величину, бóльшую $D_K$. Для модели задается процент допустимых аномальных действий, и при превышении этого порога аутентификация признается неуспешной.

### *3.1.1 Постановка эксперимента*

Для проверки пригодности вышеописанной модели был поставлен эксперимент. В нем участвовало 15 человек, каждый из которых 10 раз набирал английскую панграмму (фразу, содержащую все буквы алфавита) «The quick brown fox jumps over the lazy dog». На этих данных были построены модели, и для достижения наилучших результатов была произведена вариация порогового значения допустимых аномальных действий.

### *3.1.2 Оценка результата*

Для оценки результатов в этом и других экспериментах использовалось вычисление величин:

- False Rejection Rate (FRR) — процент нажатий пользователя, на котором система обучена, воспринятые системой как нажатия другого пользователя (ошибка 1-го рода);
- False Acceptance Rate (FAR) — процент нажатий другого пользователя, которые система определила как нажатия пользователя, на котором обучена (ошибка 2-го рода).

На эти признаки влияют два параметра:

(1) требуемая доля неаномальных нажатий для признания попытки авторизации успешной;

(2) допустимое для признания нажатия неаномальным количество стандартных отклонений продолжительности нажатия в модели.

Зависимость величин FRR и FAR от них отражают табл. 1 и 2. Легко увидеть, что наилучшие результаты этим методом достигаются при требовании попадания 75%–80% нажатий в границы двух стандартных отклонений от среднего значения, что позволяет получить результат 3,45% FAR и 8,95% FRR. Для такого сравнительно несложного подхода их следует признать достаточно высокими.

**Таблица 1** False Acceptance Rate

| Доля неаномальных нажатий | Продолжительность нажатия | | | |
|---|---|---|---|---|
| | 1,0 | 1,5 | 2,0 | 2,5 |
| 0,75 | 0% | 0,61% | 8,26% | 25,0% |
| 0,80 | 0% | 0% | 3,45% | 16,79% |
| 0,85 | 0% | 0% | 1,01% | 7,39% |
| 0,90 | 0% | 0% | 0% | 2,01% |

**Таблица 2** False Rejection Rate

| Доля неаномальных нажатий | Продолжительность нажатия | | | |
|---|---|---|---|---|
| | 1,0 | 1,5 | 2,0 | 2,5 |
| 0,75 | 96,35% | 31,41% | 1,25% | 0% |
| 0,80 | 99,43% | 52,47% | 8,95% | 0% |
| 0,85 | 100% | 84,85% | 24,16% | 4,03% |
| 0,90 | 100% | 94,97% | 64,01% | 19,59% |

## 3.2 Наблюдение за порядком нажатия и отпускания кнопок

### *3.2.1 Модель и метод*

Назовем **«обменом» («swap»)** такую пару нажатий и отпусканий, которые происходят в непрямом порядке:


Так, при обычном порядке нажатий сначала нажимается кнопка $a$, затем она отпускается, затем нажимается кнопка $b$, затем $b$ отпускается. Было замечено, что некоторые люди при наборе слов непроизвольно меняют этот порядок, нажимая кнопку $a$, затем, не отпуская ее, кнопку $b$, затем отпускают обе.

Идея метода, описываемого в этой части, заключается в учете подобных «обменов» для определения аномальности действий пользователя.

Пусть задана последовательность нажатий и отпусканий $S$, представляющая собой действия пользователя для аутентификации. Для нее можно вычислить количество таких «обменов» $x_S$ и ввести расстояние между двумя последовательностями $S_1$ и $S_2$ как $|x_{S_1} - x_{S_2}|$.

Для каждого пользователя вычислим расстояния между каждой парой его наборов, посчитаем среднее и стандартное отклонение. Эти два числа будут служить моделью для пользователя.

#### *3.2.2 Эксперимент*

Для эксперимента использовались те же данные, что и в подразд. 3.1, однако не учитывались клавиши Shift и Delete, для того чтобы не вносить излишний шум в результаты. Для обучения была использована треть собранных данных, для тестирования — две трети.

#### *3.2.3 Результаты*

Для того чтобы получить оценку аномальности, дистанция между тестируемым и пользовательским набором сравнивалась со средней дистанцией между наборами одного и того же пользователя, и при выходе значения дистанции за пределы одного стандартного отклонения попытка авторизации признавалась неудачной.

Было выяснено, что значения FAR и FRR для результатов применения этого метода сильно зависят от пар пользователей, в экспериментах проявлялись доли ошибок от 0% до 70%, что не позволяет признать этот метод подходящим для использования в одиночку. Полагается, что длина текста недостаточна для такого подхода, и предлагается усовершенствование метода путем учета того, на каких парах клавиш происходят «обмены».

### 3.3 Относительная скорость печати

Существует предположение, что на разных текстах абсолютная скорость печати пользователя варьируется в широких пределах (к примеру, осмысленный текст человек будет набирать быстрее, чем случайные символы), однако для каждой пары кнопок скорость нажатий остается примерно одинаковой. Поэтому высказывается предложение замерять скорости набора пар кнопок и использовать их в качестве модели для пользователя.

Пусть $S$ и $S'$ — векторы пар кнопок, упорядоченных по скорости набора. Пусть $S[i]$ и $S'[i]$ — положения пары кнопок $i$ в векторах $S$ и $S'$ соответственно. Тогда расстояние между этими двумя векторами вводится следующим образом: $\sum\limits_i |S[i] - S'[i]|$. Для нормировки предлагается делить расстояние на количество элементов в векторах, тогда расстояния между более полными (содержащими большее количество пар) векторами не будут резко отличаться от расстояний между векторами, содержащими меньшее число пар.

Следующие результаты были получены вариацией порогового значения расстояния между тестируемым и известным значениями вектора, необходимого для признания аутентификации удачной, на данных эксперимента, описанного в подразд. 3.1:

- расстояния между двумя векторами одного и того же пользователя составили в среднем 0,3192, расстояние между векторами разных пользователей — в среднем 0,529;
- стандартные отклонения расстояний между двумя векторами отличались незначительно.

Однако точно судить об успешности аутентификации пользователя можно только при различии в векторах, меньшем 0,3, а о неуспешности — при различии, большем 0,6.

### 3.4 Использование правой и левой клавиш Shift

Гипотетически предполагается, что при наборе текста разные люди используют правую и левую клавиши Shift по-разному. Вероятно, это можно использовать для аутентификации.

Для проверки этой гипотезы предлагается модель пользователя, состоящая из пар вида $(x, L), (y, L), (z, R)$, где первый элемент каждой пары — нажатая клавиша, а второй – элемент из множества $\{L, R\}$, соответствующий правой или левой кнопке Shift, с которым была нажата клавиша, соответствующая первому элементу.

Для эксперимента использовалась фраза, набранная каждым пользователем по 5 раз и содержащая все заглавные буквы английского алфавита: «Another Quick Brown Fox Jumps Over The Lazy Dog Yet Round Cats Eat Plain Goldfish Heartily In Maine Not Kansas Under Some Vain Zealous Xena Warrior».

По собранным данным 15 пользователей были разделены на 4 класса: 8 из них пользовались только левой клавишей Shift; четверо — только правой; двое использовали левую клавишу чаще правой; еще один — правую чаще левой.

Очевидно, только класса пользователя недостаточно, чтобы признать аутентификацию успешной. Однако попадание в чужой класс дает весомое основание отвергнуть попытку аутентификации. (Здесь не берется в расчет тот факт, что клавиатура пользователя может выйти из строя.)

### 3.5 Об одном из методов для коротких буквенных или цифровых паролей

#### *3.5.1 Предлагаемый метод и модель данных*

Одной из особенностей статической аутентификации, о которой было упомянуто в подразд. 2.1, являлся небольшой объем входных данных. В случае использования учета динамики нажатий клавиш совместно с паролем это несложно обосновать: чем короче пароль, тем легче он набирается

пользователем; с увеличением длины пароля растет и вероятность опечаток, ведущих к необходимости повторного набора. Также было упомянуто небольшое число признаков: как правило, в пароле содержится небольшое подмножество символов, буквенно-цифровых (в случае пароля для входа в систему) либо только цифровых (таким образом устроены PIN-коды в банкоматах). Из этого следует, что необходимо разрабатывать способы аутентификации с учетом ограничений по длине и разнообразию символов.

Интересным в этом свете представляется подход, описанный в статье [2]. В качестве модели авторами были взяты замеры продолжительности нажатий клавиш, однако вычислялись они тремя разными способами:

(1) **абсолютное время нажатия**. В качестве элементов вектора берутся времена удержаний клавиш и времена, когда не нажата ни одна клавиша по отдельности;

(2) **кумулятивное время удержания**. Аналогично предыдущему, однако замеряемые времена аккумулируются, что позволяет сгладить выбросы;

(3) **использование отношения задержек**. В векторе в качестве значения элемента, соответствующего нажатию, берется отношение времени удержания к продолжительности последующего за ним промежутка, когда не нажата ни одна клавиша.

В качестве алгоритма, с помощью которого осуществлялось обучение, был взят мультиклассовый линейный SVM (support vector machine) [3], так как он демонстрирует высокие результаты на данных несложной структуры.

#### *3.5.2 Постановка эксперимента*

Для участия в эксперименте были приглашены 16 человек, 8 из которых были *информированы* о проводимом эксперименте, а другие 8 — нет. Во избежание нежелательного искажения результатов непреднамеренной тренировкой 300 попыток были разбиты на 10 дней, по 30 попыток набора ежедневно.

Одна попытка представляла собой набор слова «special» в случае буквенного пароля и число «12057» в случае цифрового, что соответствовало средней длине пароля пользователя при входе в систему и длине PIN-кода. При опечатке в наборе попытка не засчитывалась, это обосновано тем, что, как правило, при наборе пароли и PIN-коды не видны пользователю и опечатка приведет к ошибке при аутентификации.

#### *3.5.3 Результаты*

Для оценки результатов применялись те же способы, что и в п. 3.1.2. Значения параметров FAR и FRR приведены в табл. 3 и 4.

В строках таблиц отражен метод замера времени, в столбцах — результаты по данным, полученным от информированных и неинформированных пользователей.

Несложно заметить, что процент ошибок сильно (в разы) зависит от информированности пользователей и метод замера отношений несколько более эффективен, нежели два прочих.

**Таблица 3** Буквенный пароль

| Метод замера | Неинформированные пользователи | | Информированные пользователи | |
|---|---|---|---|---|
| | FAR | FRR | FAR | FRR |
| Абсолютный | 6,82% | 12,30% | 1,59% | 3,82% |
| Кумулятивный | 5,92% | 11,43% | 1,32% | 3,15% |
| Отношений | 6,73% | 11,69% | 0,91% | 2,31% |

**Таблица 4** Цифровой пароль

| Метод замера | Неинформированные пользователи | | Информированные пользователи | |
|---|---|---|---|---|
| | FAR | FRR | FAR | FRR |
| Абсолютный | 5,67% | 10,36% | 1,75% | 3,21% |
| Кумулятивный | 4,93% | 9,69% | 1,31% | 2,58% |
| Отношений | 5,10% | 9,92% | 0,99% | 1,92% |

# 4 Обзор существующих методов динамической аутентификации

## 4.1 Слияние классификаторов

В качестве события $A$ будем рассматривать некоторую порцию данных, например набор одного абзаца или строки. Назовем ее сессией (не путать с сессией работы с системой). Весь процесс аутентификации будет рассмотрен именно для сессий, а не для отдельных нажатий клавиш.

#### *4.1.1 Формат данных*

В данном методе в качестве регистрируемых данных рассматриваются задержки между событиями, производимыми клавиатурой. Событие представляет собой нажатие или отпускание клавиши. Очевидно, что для одной и той же пары нажатий можно использовать различные интервалы времени:

– PP (Press–Press): время между двумя последовательными нажатиями клавиш;

– PR (Press–Release): время, в течение которого клавиша была нажата;
– RP (Release–Press): время между отпусканием предыдущей и нажатием следующей клавиши;
– RR (Release–Release): время между двумя последовательными отпусканиями клавиш.

Все остальные характеристики, такие как сила нажатия на клавишу и пр., в данном исследовании не рассматриваются, так как могут быть получены только при помощи дополнительного оборудования. Для анализа будем использовать совокупность временны́х значений PP, PR, RP, RR для вводимой с клавиатуры последовательности.

#### *4.1.2 Определение по расстоянию до среднего*

В случае аутентификации часто используется метод [4]. Для формирования профиля пользователя рассчитывается среднее значение $\mu$ и стандартное отклонение $\sigma$ для всех типов событий. Одно событие считается корректным, если его значение отличается от среднего не более чем на половину отклонения. Иными словами, $|t-\mu| < \alpha\sigma$. Сессия считается корректной, если доля корректных событий в ней не менее некоторого порога $\beta$.

Можно доработать этот метод, адаптируя параметры под каждого конкретного пользователя [5]. Например, изначально установить порог $\beta = 1$ и далее понижать его на заданное небольшое число, пока считаются корректными все события из тренировочного набора.

Также вместо того, чтобы непосредственно сравнивать время со средним, можно использовать взвешенную оценку. Для этого каждому событию сопоставим оценку (лежащую в интервале $[0,1]$) $\mathrm{score} = \exp(|t-\mu|/\sigma)$. В качестве оценки для сессии возьмем среднее по оценкам, входящим в нее. Это среднее далее и будем сравнивать с порогом $\beta$.

#### *4.1.3 Определение по ритму набора*

Как и в музыке, где ритм определяется как относительная длительность нот (половина ноты, четверть ноты и т. д.), предлагается рассмотреть ритм, с которым пользователь набирает текст [5]. Основная идея здесь заключается в том, чтобы разделить численные временны́е значения на несколько дискретных классов. Для этого можно использовать пороговые значения:

– $t > 200$: класс 1;
– $100 < t < 200$: класс 2;
– $70 < t < 100$: класс 3;
– $30 < t < 70$: класс 4;
– $t > 30$: класс 5.

Недостаток такого подхода в том, что при использовании статических порогов не учитывается средняя скорость набора текста пользователем. Для решения этой проблемы можно классифицировать время внутри одной сессии в сравнении с остальными интервалами в ней.

– $1/(10)$ самых медленных: класс 1;
– $1/3$ самых медленных: класс 2;
– $2/3$ самых медленных: класс 3;
– $3/4$ самых медленных: класс 4;
– $1/4$ самых быстрых: класс 5.

В качестве профиля пользователя каждому нажатию сопоставляем вектор классов, к которым отнесены соответствующие временны́е интервалы. Далее над этими векторами вводим дистанцию как сумму разниц между номерами классов в векторах. Далее, сравнивая эту сумму с порогом, получаем критерий корректности пользователя.

#### *4.1.4 Определение по ранжированию времен*

Данный метод используется только для статической аутентификации [6]. Для каждого события считается его ранг (порядковый номер в отсортированной последовательности тех же значений). В качестве профиля пользователя вычисляется среднее по рангам для каждого из нажатий (по тестовой выборке). Для того чтобы оценить сессию, используется коэффициент Спирмана, который считается по формуле:

$$r_{\mathrm{Sp}} = 1 - \frac{6\sum_{i=1}^{n}\left(r_i^1 - r_i^2\right)}{n(n^2-1)},$$

где $r_i^1$ – ранг $i$-го нажатия в сессии 1, а $n$ — количество нажатий в сессии.

#### *4.1.5 Слияние методов*

Пусть для каждого пользователя имеется три классификатора, работающих на одних и тех же данных, но оценивающих их различные характеристики. Для комбинирования их в одном подходе воспользуемся правилами слияния, описанными в [7].

Итак, имеется три классификатора, которые решают проблему о принадлежности к одному из двух классов (корректный пользователь / нарушитель), возвращающих оценку от 0 до 1. Проблема в том, что оценки каждого из классификаторов не похожи и по-разному распределены для одних и тех же наборов данных. В связи с этим неэффективно было бы применять любую схему голосования («хотя бы

один», «большинство», «все»). Для этого существует несколько методов нормализации:

– нормализация по максимуму:

$$\text{score}' = \frac{\text{score}}{\text{scoreMax}};$$

– нормализация по максимальной разнице:

$$\text{score}' = \frac{\text{score} - \text{scoreMin}}{\text{scoreMax} - \text{scoreMin}};$$

– Z-мера:

$$\text{score}' = \frac{|\text{score} - \overline{\text{score}}|}{\sigma_{\text{score}}}.$$

Стоит отметить, что первые два метода нормализации могут привести к получению значений больше 1 или меньше 0, однако в данном случае это будет обозначать векторы, крайне близкие или отдаленные от профиля пользователя.

Все эти оценки требуют предварительного знания о самой выборке, а также дополнительного расчета (особенно в случае Z-меры), поэтому использовать их в реальном времени весьма затруднительно. Можно использовать методы слияния, описанные в [7]. Для начала сведем задачу к бинарной классификации с ограничением

$$P(\text{user}) = 1 - P(\text{impostor}).$$

Так как нет информации о вероятности, положим

$$P(\text{user}|i) = \text{Score}^i,$$

где $\text{Score}^i$ — оценка $i$-го классификатора. Далее можно использовать следующие операторы слияния:

– максимум и минимум:

$$\text{Score} = \max_i(\text{Score}^i);$$

– медиана:

$$\text{Score} = \text{MEDIAN}(\text{Score}^i);$$

– правило произведения:

$$\text{Score} = \prod_i \text{Score}^i;$$

– правило суммы:

$$\text{Score} = \sum_i \text{Score}^i.$$

#### *4.1.6 Результаты*

Для сравнения методов использовались три меры оценки:

(1) FAR — процент некорректных пользователей, которые были приняты классификатором (ошибка 2-го рода);

(2) FRR — процент корректных пользователей, которые были отвергнуты классификатором (ошибка 1-го рода);

(3) EER (Equal Error Rate) — точка, в которой FAR = FRR.

**Таблица 5** Средние ошибки

| Метод | FAR | FRR | ERR |
|---|---|---|---|
| Расстояние до среднего | 4,39% | 4,81% | 4,60% |
| Расстояние до среднего (взвешенное) | 3,62% | 3,61% | 3,62% |
| Ранжирование по Спирману | 3,56% | 3,62% | 3,59% |
| Ритм набора (по порогам) | 3,47% | 3,39% | 3,43% |
| Ритм набора (пропорциональный) | 3,55% | 4,02% | 3,79% |

**Таблица 6** Применение нормализации

| Нормализация | FAR | FRR | ERR |
|---|---|---|---|
| По максимуму | 1,75% | 2,46% | 2,11% |
| По максимальной разнице | 2,00% | 2,00% | 2,00% |
| Z-мера | 1,81% | 1,69% | 1,75% |

**Таблица 7** Результаты слияния

| Метод | FAR | FRR | ERR |
|---|---|---|---|
| Голосование («все») | 1,17% | 7,92% | 4,55% |
| Голосование («большинство») | 2,39% | 2,15% | 2,27% |
| Голосование («хотя бы один») | 7,60% | 0,54% | 4,07% |
| Правило произведения | 2,00% | 2,00% | 2,00% |
| Правило максимума | 3,62% | 3,61% | 3,62% |
| Правило минимума | 3,62% | 3,62% | 3,62% |
| Правило медианы | 3,34% | 3,39% | 3,37% |
| Правило суммы | 1,81% | 1,69% | 1,75% |

Значения этих мер оценки на разных методах приведены в табл. 5 и 6. Как видно из табл. 7, наилучший результат для данного метода достигается при применении правила суммы для результатов трех исходных классификаторов.

## 4.2 Кластеризация методом Partition Around Medoids

Сначала рассмотрим алгоритм классификации пользователей, основанный на кластеризации и представленный в [8].

#### *4.2.1 Сбор данных и модель представления*

В данной статье был использован набор данных Si6 [9], который состоял из 66 сессий набора 62 различных пользователей.

Каждая сессия набора состояла из 15 предложений. Для каждой сессии фиксировалось набираемое предложение, время нажатия (или отпускания) клавиши с точностью до миллисекунд, сама клавиша и тип действия (нажатие или отпускание).

В эксперименте учитывались только завершенные сессии. Для трех пользователей, которые набрали несколько сессий, была выбрана только одна. Также не учитывались ошибочные предложения. После исключения осталось 54 сессии, каждая из которых состояла из 15 предложений.

Для каждого предложения каждой сессии каждого пользователя был вычислен вектор времени диграфов (время между двумя последовательными нажатиями на клавиши). Диграфы, содержащие непечатные символы, были исключены. Рассматривались оставшиеся 88 946 времен диграфов, соответствующих 411 уникальным последовательностям из двух клавиш.

Были проведены три различных эксперимента с разной группировкой предложений пользователя в подсессии:

(1) 3 подсессии, каждая состояла из 5 предложений;

(2) 5 подсессий, каждая состояла из 3 предложений;

(3) 15 подсессий, каждая состояла из 1 предложения.

Каждая подсессия характеризовалась вектором средних значений всех различных диграфов, отсортированных по убыванию числа появлений у всех пользователей.

Количество кластеров для классификации (параметр $k$) было выбрано по алгоритму *силуэта*, описанному в [10].

#### *4.2.2 Фильтрация*

Фильтрация играет значимую роль в процессе интеллектуального анализа данных. Были использованы два уровня фильтрации:

(1) при грубой фильтрации были исключены:

   (а) диграфы со временем менее 10 и более 750 мс;

   (б) диграфы, время которых отличалось от среднего времени всех диграфов больше чем на стандартное отклонение времени всех диграфов, т. е. $140 \pm 113$ мс $\approx$ (25 мс, 250 мс).

   Таким образом было исключено около 17% диграфов;

(2) на этапе более тонкой фильтрации исключались:

   (а) диграфы, встречающиеся только один раз;

   (б) диграфы, у которых стандартное отклонение больше удвоенного среднего стандартного отклонения всех диграфов данной подсессии;

   (в) из одинаковых диграфов в подсессии исключались те, которые имели время, отличающееся от среднего в данной выборке больше чем на удвоенное стандартное отклонение.

#### *4.2.3 Результаты*

При оценке качества определение в один кластер двух подсессий, принадлежащих одному пользователю, являлось верной классификацией (TP — true positive). Ложно-положительным (FP — false positive) срабатыванием считалось определение двух подсессий различных пользователей в один кластер. Ложно-отрицательным (FN — false negative) — определение двух подсессий одного пользователя в разные кластеры.

Значение полноты (Precision) было принято равным $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$; точности (Recall) — $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$; $F$-мера:

$$F = 2\frac{\text{Presicion} \cdot \text{Recall}}{\text{Presicion} + \text{Recall}}.$$

Лучшие значения $F$-меры составили: для разделения сессий на 3 подсессии — 0,9, на 5 подсессий — 0,78, на 15 подсессий — 0,26.

### 4.3 Относительная скорость печати

Теперь рассмотрим метод, описанный в [11]. Центральным понятием здесь является $n$-граф — время между первым и последним нажатием серии из $n$ клавиш.

В нем описывается использование как относительных метрик различия между сессиями печати, которые учитывают только то, какие $n$-графы набираются быстрее других, так и абсолютных, сравнивающих время набора.

#### *4.3.1 Относительные метрики ($R$-метрики)*

Для массива $V$ из $k$ чисел можно вычислить степень неупорядоченности, которая является суммой расстояний между элементом $V$ и соответствующим элементом упорядоченного массива $V'$. Так, для массива $A = [2, 3, 1, 4, 5]$ степень неупорядоченности равна $1 + 1 + 2 + 0 + 0 = 4$.

Получим максимальную степень неупорядоченности массива из $k$ чисел:

$$D_k = \begin{cases} \dfrac{k^2}{2}\,, & k \text{ — четное}\,; \\ \dfrac{k^2-1}{2}\,, & k \text{ — нечетное}\,. \end{cases}$$

Это позволит нормировать неупорядоченность. Так, для массива $A$ из примера нормированная неупорядоченность будет равна:

$$\frac{1+1+2+0+0}{(5^2-1)/2} = \frac{8}{12} = 0{,}6666\,.$$

Образец печати пользователя $S$ представляет собой массив средних времен набора $n$-графов, встречающихся в тексте, отсортированный по возрастанию. Таким образом, вводится расстояние между двумя образцами печати $S_1$ и $S_2$. Если у $S_1$ и $S_2$ $k$ общих $n$-графов, то расстояние $R_n(S_1, S_2)$ равно сумме расстояний между общими $n$-графами, нормированному на $D_k$. Те $n$-графы, которые не принадлежат пересечению $S_1$ и $S_2$, просто игнорируются.

Заметим, что для небольших наборов данных у двух образцов может не оказаться общих $n$-графов, что сделает невозможным вычисление $R_n$.

Можно предположить, что вычисление относительной метрики между двумя образцами для диграфов, триграфов или иных $n$-графов способно предоставить различные аспекты информации о различии ритмов печати.

Если у образцов $S_1$ и $S_2$ $N$ общих $n$-графов и $M$ общих $m$-графов ($N > M$), вводится кумулятивная относительная метрика:

$$R_{n,m} = R_n(S_1, S_2) + R_m(S_1, S_2)\frac{M}{N}\,.$$

Аналогично можно расширить метрику для применения к большему количеству различных $n$-графов:

$$R_{n,m} = R_n(S_1, S_2) + R_m(S_1, S_2)\frac{M}{N} + R_p(S_1, S_2)\frac{P}{N}\,.$$

#### *4.3.2 Абсолютные метрики ($A$-метрики)*

К сожалению, у $R$-метрик есть существенный недостаток. Если среднее время каждого $n$-графа в образце $S_2$ в два раза больше, чем в образце $S_1$, то $R_n(S_1, S_2) = 0$. Таким образом, относительная метрика не может отличить двух пользователей, имеющих очень похожие ритмы печати, пусть даже один из них печатает намного быстрее другого.

В отличие от $R$-метрик, $A$-метрики учитывают абсолютное время набора $n$-графов. Пусть $G_{S_1,d_1}$ и $G_{S_2,d_2}$ — это один и тот же $n$-граф, присутствующий в $S_1$ со временем $d_1$ и в $S_2$ со временем $d_2$. *Похожими* называются $n$-графы, для которых выполняется соотношение

$$1 < \frac{\max(d_1, d_2)}{\min(d_1, d_2)} \le t\,,$$

где $t$ — некоторая константа, большая единицы.

Введем метрику

$$A_n^t(S_1, S_2) = 1 - \frac{\text{количество } \textit{похожих}\ n\text{-графов}}{\text{общее количество } n\text{-графов}}\,.$$

Таким образом, расстояние между образцами, у которых все $n$-графы похожи, будет равно 0. Между образцами, у которых нет общих $n$-графов, — 1. Кумулятивные метрики вводятся аналогично $R$-метрике:

$$A_{n,m}^t = A_n^t(S_1, S_2) + A_m^t(S_1, S_2)\frac{M}{N}\,;$$

$$A_{n,m}^t = A_n^t(S_1, S_2) + A_m^t(S_1, S_2)\frac{M}{N} + A_p^t(S_1, S_2)\frac{P}{N}\,.$$

Стоит заметить, что во всех ниже описываемых экспериментах принималось $t = 1{,}25$.

#### *4.3.3 Сбор данных*

Сорок человек предоставили по 15 сессий печати. Они выполняли роль легитимных пользователей системы. Еще 165 человек предоставили по одной сессии. Эти образцы использовались для имитации атаки на систему.

Перерыв между сессиями был не менее одного дня. Каждая сессия представляла собой свободный текст длиной около 700–900 символов.

У всех участников эксперимента итальянский является родным языком.

#### *4.3.4 Аутентификация*

В оригинальной статье рассматриваются эксперименты по классификации, идентификации и аутентификации. В рамках данного обзора будет описан только эксперимент по аутентификации.

Пусть рассматриваются пользователи $A$, $B$, $C, \ldots$, которые представлены образцами печати $A_1, \ldots, A_n$, $B_1, \ldots, B_k$ и т. д. Средним расстоянием

неизвестного образца $X$ от пользователя $A$ назовем

$$\mathrm{md}(A,X) = \\ = \frac{1}{n}\left(d(A_1,X) + d(A_2,X) + \cdots + d(A_n,X)\right),$$

где $d(A_i,X)$ — расстояние между двумя образцами.

Среднее расстояние между образцами пользователя $A$ обозначим как $m(A)$.

Неизвестный образец $X$ считается принадлежащим шаблону пользователя $A$ в случае выполнения следующих условий:

- $\mathrm{md}(A,X) < \mathrm{md}(B,X)$ для любого другого легитимного пользователя $B$;
- $\mathrm{md}(A,X) < m(A)$ или $\mathrm{md}(A,X) - m(A) < \mathrm{md}(B,X) - \mathrm{md}(A,X)$ для любого другого легитимного пользователя $B$.

Каждый образец $S$ легитимного пользователя $U$ был использован как новый и неизвестный образец. При корректной работе классификатора он должен быть отнесен к пользователю $U$, представленному оставшимися 14 образцами. Также $S$ использовался для попытки аутентификации под видом каждого из оставшихся 39 пользователей. Когда образец $S$ пользователя $U$ используется для аутентификации под видом пользователя $U'$, пользователь $U$ временно изымается из модели. Таким образом, нарушитель всегда был неизвестен.

Каждый из 165 образцов нарушителей был использован для попытки аутентификации под видом каждого легитимного пользователя. Получается 600 попыток аутентификации правильным пользователем и 450 000 атак $40 + 165 = 205$ нарушителями ($600 \cdot 39 \cdot 15 + 165 \cdot 40 \cdot 15 = 450\,000$).

#### *4.3.5 Результаты*

В табл. 8 представлены результаты первого эксперимента. В нем были использованы одиночные метрики, перечисленные в первом столбце. В табл. 8 FP count — количество аутентифицированных атак (из 450 000); FN count — количество отвергнутых легитимных аутентификаций (из 600); FP rate и FN rate — отношение FP count и FN rate к 450 000 и 600 соответственно.

**Таблица 8** Одиночные метрики

| Метрики | FP count | FN count | FP rate, % | FN rate, % |
|---|---|---|---|---|
| $R_2$ | 563 | 50 | 0,13 | 8,33 |
| $R_{2,3}$ | 324 | 32 | 0,07 | 5,33 |
| $R_{2,4}$ | 259 | 41 | 0,06 | 6,83 |
| $R_{2,3,4}$ | 199 | 41 | 0,04 | 6,83 |
| $A_2$ | 590 | 92 | 0,13 | 15,3 |
| $A_{2,3}$ | 335 | 80 | 0,07 | 13,3 |
| $A_{2,4}$ | 366 | 84 | 0,08 | 14,0 |
| $A_{2,3,4}$ | 331 | 79 | 0,07 | 13,2 |

**Таблица 9** Сумма метрик

| Метрики | FP count | FN count | FP rate, % | FN rate, % |
|---|---|---|---|---|
| $R_2 + A_2$ | 360 | 36 | 0,08 | 6,0 |
| $R_2 + A_{2,3}$ | 272 | 34 | 0,06 | 5,667 |
| $R_2 + A_{2,4}$ | 260 | 37 | 0,057 | 6,167 |
| $R_2 + A_{2,3,4}$ | 237 | 41 | 0,052 | 6,833 |
| $R_{2,3,4} + A_2$ | 124 | 19 | 0,028 | 3,167 |
| $R_{2,3,4} + A_{2,3}$ | 78 | 23 | 0,017 | 3,833 |
| $R_{2,3,4} + A_{2,4}$ | 95 | 22 | 0,021 | 3,667 |
| $R_{2,3,4} + A_{2,3,4}$ | 131 | 23 | 0,029 | 3,833 |

**Таблица 10** Последовательные метрики

| Метрики | FP count | FN count | FP rate, % | FN rate, % |
|---|---|---|---|---|
| $\{R_2 + A_2,\ R_2 + A_{2,3},\ R_2 + A_{2,4}.\ R_2 + A_{2,3,4}\}$ | 83 | 55 | 0,018 | 9,167 |
| $\{R_{2,3} + A_2,\ R_{2,3} + A_{2,3},\ R_{2,3} + A_{2,4},\ R_{2,3} + A_{2,3,4}\}$ | 74 | 38 | 0,016 | 6,333 |
| $\{R_{2,3,4} + A_2,\ R_{2,3,4} + A_{2,3},\ R_{2,3,4} + A_{2,4},\ R_{2,3,4} + A_{2,3,4}\}$ | 22 | 29 | 0,005 | 4,833 |

Для второго эксперимента, результаты которого отражены в табл. 9, метриками служили суммы известных метрик.

В третьем эксперименте (табл. 10) образец считался аутентифицированным, если он признавался аутентифицированным каждой метрикой последовательности.

## 4.4 Метод наказания-поощрения (*penalty-reward*)

Интересный метод был предложен в [12]. Он основан на динамическом изменении уровня недоверия к пользователю, что позволяет оперативно реагировать на подмену пользователя.

### *4.4.1 Функция наказания и поощрения*

Во время набора пользователь характеризуется уровнем доверия $C$. В начале сессии он принимается равным 0. При каждом нажатии $C$ изменяется в зависимости от информации в шаблоне пользователя. Если ритм нажатия хорошо подходит к шаблону, пользователь поощряется уменьшением $C$. В противном случае — наказывается увеличением.

Пока $C$ остается ниже некоторого порога, есть уверенность в том, что это тот пользователь, за которого он себя выдает. Если же значение $C$ велико, то, скорее всего, системе нужно предпринять действия по уточнению личности пользователя.

Значение $C$ не должно становиться отрицательным. Если этого не учитывать, то во время работы легитимного пользователя $C$ станет очень маленьким, что даст нарушителю много времени до превышения $C$ заданного порога. Нужно решить, как должно меняться значение уровня доверия в случае, если очередная клавиша (комбинация клавиш) не входят в шаблон пользователя. Авторами предлагается увеличивать $C$ на небольшую заданную константу.

Таким образом, можно ввести следующую функцию:

$$C = \begin{cases} 0\,, & \text{начало сессии}\,; \\ \max(C - R, 0)\,, & d \le T\,; \\ C + d - T\,, & d > T\,; \\ C + \alpha\,, & \text{клавиша не из шаблона}\,. \end{cases}$$

Здесь $R$ – величина поощрения пользователя за нажатие, соответствующее шаблону; $d$ — расстояние от очередного нажатия до шаблона пользователя; $T$ — доверительный порог расстояния; $\alpha$ — наказание за клавишу, не принадлежащую шаблону.

### *4.4.2 Сбор данных и модель представления*

В эксперименте участвовало 25 пользователей, предоставляя информацию о нажатиях в течение 6–15 дней своей обычной работы за компьютером.

Время между нажатием и отпусканием клавиши назовем *удержанием*. Время между отпусканием одной клавиши и нажатием следующей назовем *задержкой*.

Шаблон пользователя представляет собой математическое ожидание ($\mu$) и стандартное отклонение ($\sigma$) удержания и задержки нажатых клавиш и комбинаций клавиш. Решение о включении клавиши (комбинации клавиш) в шаблон принималось на основе числа вхождений $N$ в наборе и отношения $\mu$ и $\sigma$: $N$ должно быть выше некоторого порога, $\mu/\sigma$ — ниже.

Расстояние между новой клавишей и шаблоном введено как

$$d = d((\mu, \sigma), t) = \left| \frac{t - \mu}{\sigma} \right| ,$$

где $t$ — время удержания или задержки.

Расстояние между комбинацией клавиш $k_1k_2$ и шаблоном:

$$d = \frac{1}{3} \left( \left| \frac{t_{k_1} - \mu_{k_1}}{\sigma_{k_1}} \right| + \left| \frac{t_{k_1k_2} - \mu_{k_1k_2}}{\sigma_{k_1k_2}} \right| + \left| \frac{t_{k_2} - \mu_{k_2}}{\sigma_{k_2}} \right| \right).$$

Необходимо определить максимальное значение $C$, после которого можно считать пользователя нарушителем — $T_{\text{action}}$. Это значение будет своим для каждого пользователя. Оно будет равно максимальному значению $C$, полученному при применении шаблона пользователя к обучающим данным. Это позволит свести к минимуму ложно-отрицательные срабатывания.

### *4.4.3 Результаты*

Для каждой пары различных пользователей $i$ и $j$ шаблон $i$ применялся к данным $j$. Измерялось среднее количество нажатий, после которого значение $C$ превышало порог $T^i_{\text{action}}$.

Требовалось от 79 до 348 нажатий для определения нарушителя. В среднем нарушитель определялся после 265 нажатий.

# 5 Предлагаемый подход

## 5.1 Представление данных

Среди рассмотренных выше способов лишь подход, использующий $n$-графы, учитывает в единице данных события, происходившие на протяжении некоторого времени. Тем не менее очевидно, что

параметры, относящиеся только к одному нажатию, несут в себе меньше информации, нежели параметры, описывающие их серию. Поэтому предлагается новый подход, позволяющий эффективно сохранять недавнюю активность пользователя при работе с клавиатурой.

Предлагаемое представление данных основывается на отображении

$$\varphi : (A, t_d, t_h) \to H, A \in \Omega\,,$$

где $\Omega$ – некоторый конечный алфавит действий пользователя; $t_d$ — время, в которое была нажата клавиша; $t_h$ — время, в течение которого клавиша удерживалась нажатой; $H$ — вектор признаков. Это отображение должно оказывать большее влияние на анализ события $A^{\text{next}}$ в зависимости от следующих факторов:

– недавняя активность пользователя;

– частота совершения действия;

– протяженность действия по времени.

Для построения такого отображения была использована теория потенциальных функций [13]. Пусть каждое возможное событие $A_i \in \Omega$ имеет свой потенциал в момент $t$. Он убывает в зависимости от времени, прошедшего с момента совершения действия. Этот процесс характеризуется функцией $\mathrm{Pf} : \mathrm{Time} \times \mathrm{Time} \to \mathbb{R}$. Если последовательность содержит два или более событий $A_i$, то их потенциалы суммируются.

Таким образом, можно определить отображение последовательности действий пользователя в $L$-мерный вещественный вектор, где $L = |\Omega|$, в виде:

$$\varphi(H(U), t) = \left( \sum_{\substack{(A,t_m)\in H(U) \\ t>t_m}} Pf(t, t_m) \right)_{A\in\Omega}.$$

Согласно этой формуле, активность пользователя в каждый конкретный момент времени $t$ может быть определена как множество из $L = |\Omega|$ потенциалов $\varphi_A(H(U), t)$. Такой подход учитывает как частоту предыдущих действий, так и время, в которое текущее действие было совершено.

Был выбран класс радиально-базисных функций, поскольку эти функции обладают необходимым в данном случае свойством: они зависят от интервалов между действиями, но не зависят от абсолютного времени их совершения. Кроме того, эти функции легко параметризовать для того, чтобы добиться эффективности в задачах аутентификации и идентификации.

В качестве потенциальной радиально-базисной функции была выбрана экспоненциальная функция $\mathrm{Pf}(x, y) = e^{-\sigma||x-y||}$, где $\sigma$ — коэффициент затухания, отвечающий за то, как быстро потенциал будет убывать. Кроме того, для конечной последовательности действий эта функция может быть рассчитана рекурсивно (полагая $\varphi_A(0) = 0$):

$$\varphi_A(t_n) = \begin{cases} \varphi_A(t_{n-1})e^{-\sigma||t_n - t_{n-1}||}\,, & A \neq A_n; \\ e^{-\alpha}\,, & A = A_n\,. \end{cases}$$

Для достижения максимальной эффективности коэффициенты $\sigma$ и $\alpha$ могут быть подобраны, учитывая специфику конкретной задачи. Коэффициент $\alpha$ отвечает за влияние времени нажатия на значение потенциала.

Для того чтобы избежать зашумленности данных низкими значениями, можно ввести константный порог $\varepsilon$ следующим образом:

$$\hat{\varphi}_A(t_n) = \begin{cases} \varphi_A(t_n)\,, & \varphi_A(t_n) \geq \varepsilon\,; \\ 0\,, & \varphi_A(t_n) < \varepsilon\,. \end{cases}$$

Текущая активность пользователя описывается последовательностью $n$-мерных векторов (где $n = |\Omega|$), содержащих потенциалы $\hat{\varphi}_A$ для каждого действия $A$.

В качестве примера рассмотрим набор слова Hello и получаемые при этом векторы. Возьмем упрощенный случай: $\Omega$ ограничим буквами H, E, L, O. В таком случае получаемые векторы будут четырехмерными и могут иметь вид, отраженный в табл. 11.

На рис. 1 видно, что элементы вектора представляют собой значения потенциальной функции соответствующего действия в моменты нажатий.

Таким образом, один вектор содержит достаточное количество информации, чтобы быть правильно распознанным с высокой долей вероятности с помощью алгоритмов машинного обучения, и при этом генерируется на каждое нажатие, что позволяет увеличить объем данных для обучения и распознавания.

**Таблица 11** Получаемые векторы при наборе слова Hello

| Событие | Состояние вектора | | | |
|---|---|---|---|---|
| | «H» | «E» | «L» | «O» |
| Начальное состояние | 0 | 0 | 0 | 0 |
| Нажата «H» | 1 | 0 | 0 | 0 |
| Нажата «E» | 0,82 | 0 | 0 | 0 |
| Нажата «L» | 0,53 | 0,64 | 1 | 0 |
| Нажата «L» | 0,34 | 0,41 | 1 | 0 |
| Нажата «O» | 0,23 | 0,30 | 0,80 | 1 |

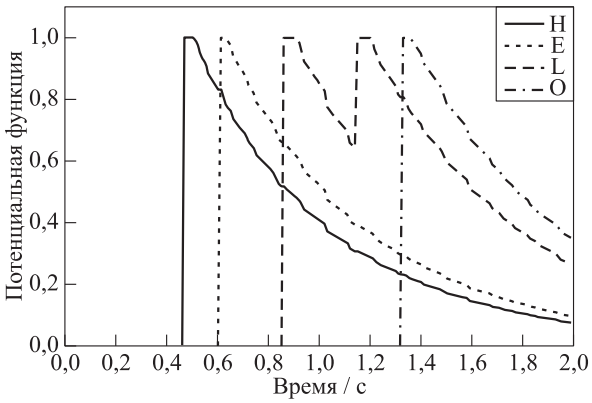



**Рис. 1** Набор слова «Hello»

## 5.2 Сбор данных

Для проверки качества предлагаемой модели представления был использован уже описанный выше набор данных Si6 [9], состоящий из 55 пользователей.

Кроме этого, в рамках исследовательской работы был проведен независимый сбор данных. Сорока испытуемым было предложено установить программу-агент, которая в фоновом режиме собирала информацию о нажатиях во время их обычной работы (преимущественно с текстовым редактором и веб-браузером). Весь эксперимент был разделен на 2 дня, в каждый из которых условия или оборудование для испытуемого оставались неизменными. Таким образом, для пользователя регистрировались две сессии длительностью 3 ч в первый день и 6 ч во второй. В среднем на одного испытуемого приходилось порядка 24 тыс. нажатий.

## 5.3 Статическая аутентификация с предлагаемой моделью поведения

Для проверки нового представления данных был проведен эксперимент по статической аутентификации. Все исходные данные обрабатывались потенциальным функтором, и для анализа использовались только векторы значений потенциальной функции, полученные в результате.

Для каждого пользователя все данные в наборе разделялись на «свои» и «чужие» (по принадлежности вектора этому пользователю). В качестве тренировочного набора бралось по 1500 векторов, полученных случайной выборкой из «своих» и «чужих» данных (всего 3000 векторов), в качестве тестовой выборки из каждого набора бралось по 1000 векторов (всего 2000).

На тренировочном наборе было обучено 2 классификатора:

(1) **дерево решений.** В качестве реализации был выбран алгоритм See5/C5.0 [14];

(2) **случайный лес** (ансамбль деревьев решений). Была взята реализация алгоритма из [15]. В данном эксперименте использовался ансамбль из 500 деревьев решений.

Далее каждому классификатору на вход подавались векторы из тестового набора, для каждого из которых он выдавал вероятность принадлежности данного события классу «свой».

На основании этого алгоритма было проведено 2 эксперимента:

(1) первый эксперимент был проведен на наборе данных Si6 [9]. ROC-кривые (Receiver Operating Characteristics) для этого набора представлены

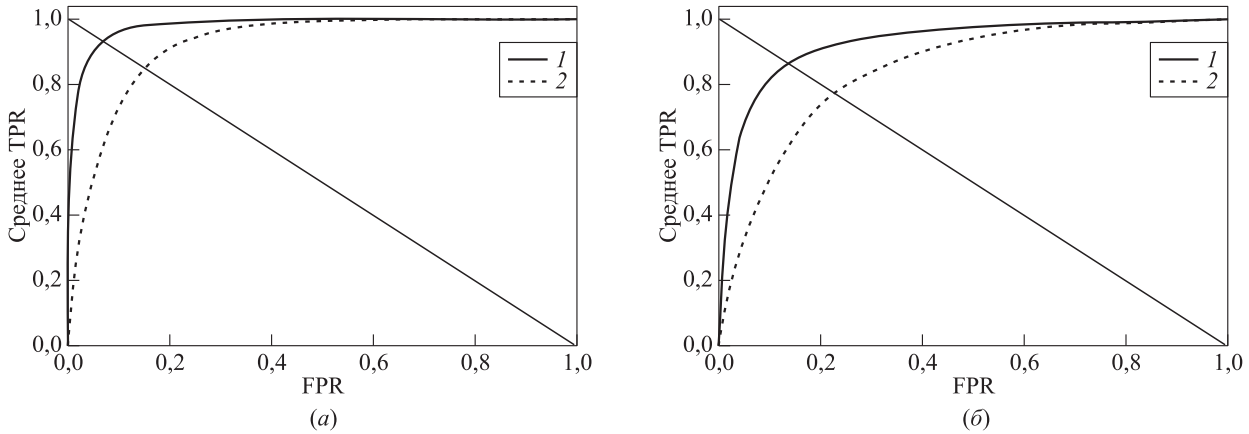



**Рис. 2** ROC-кривые (TPR — true positive rate; FPR — false positive rate) для набора данных Si6 (*а*) и для данных, собранных авторами (*б*): *1* — случайный лес; *2* — дерево решений (алгоритм C5.0)

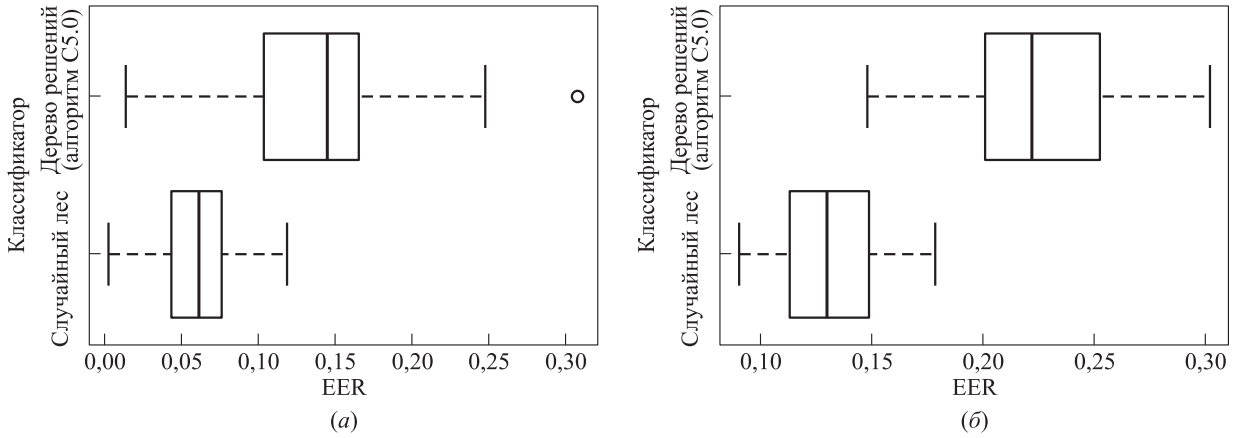



**Рис. 3** Распределение EER для данных Si6 (*а*) и для данных, собранных авторами (*б*)

на рис. 2, *а*. Как видно из распределений на рис. 3, *а*, наилучший EER достигался при использовании ансамбля деревьев решений и составлял в среднем 6%. В этом подходе $F$-мера оказалась равной 0,94;

(2) второй эксперимент был проведен на данных, собранных авторами статьи (рис. 2, *б*). Здесь показатели несколько хуже и, как видно из распределений на рис. 3, *б*, EER в среднем составляет 13%.

## 5.4 Непрерывная динамическая аутентификация с предлагаемой моделью представления

Для проверки качества представления данных был проведен эксперимент по непрерывной динамической аутентификации, использующий функцию наказания-поощрения.

В качестве алгоритма интеллектуального анализа данных были использованы деревья решений в реализации C5.0 [14].

Функция наказания-поощрения, записанная выше, применительно к данному представлению (отсутствие данных о нажатии какой-либо клавиши не является особым случаем, который надо рассматривать отдельно) принимает следующий вид:

$$C = \begin{cases} 0\,, & \text{начало сессии}\,; \\ \max(C - R, 0)\,, & d \le T\,; \\ C + d - T\,, & d > T\,, \end{cases}$$

где $d$ — вероятность того, что очередной вектор соответствует шаблону пользователя, выданная классификатором.

### *5.4.1 Постановка эксперимента*

В рамках эксперимента моделировалась ситуация, при которой каждый пользователь делал попытку аутентификации под всеми другими пользователями.

В каждом случае классификатор обучался на 2000 векторов пользователя, под видом которого происходила аутентификация, и на 2000 случайных векторов всех других пользователей (кроме того, который производил попытку аутентификации). Таким образом, атакующий пользователь никогда не принимал участия в обучении классификатора.

Валидационный набор представлял собой 1000 дополнительных векторов атакуемого пользователя. Порог уровня доверия, при превышении которого пользователь считался нарушителем, был взят в 10 раз выше, чем максимальный уровень доверия на валидационном наборе.

На рис. 4 показаны уровни доверия для валидационного набора легитимного пользователя и тестового набора атакующего пользователя соответственно. Обратите внимание на различие в единицах измерения порога. В данном примере атака была определена на 212-м нажатии. Заметим, что уровень доверия легитимного пользователя не превосходит 1,4 на протяжении 1000 нажатий. Доверие тестового набора с определенного момента только увеличивается.

Таким образом, для каждого из 55 пользователей было получено 54 номера нажатия, на которых уровень доверия превышал указанный порог. Для оценки качества алгоритма был взят средний номер нажатия в каждой выборке.

Аналогичным образом был проведен эксперимент и на данных, собранных авторами. Результаты показаны на рис. 5. В половине случаев наруши-

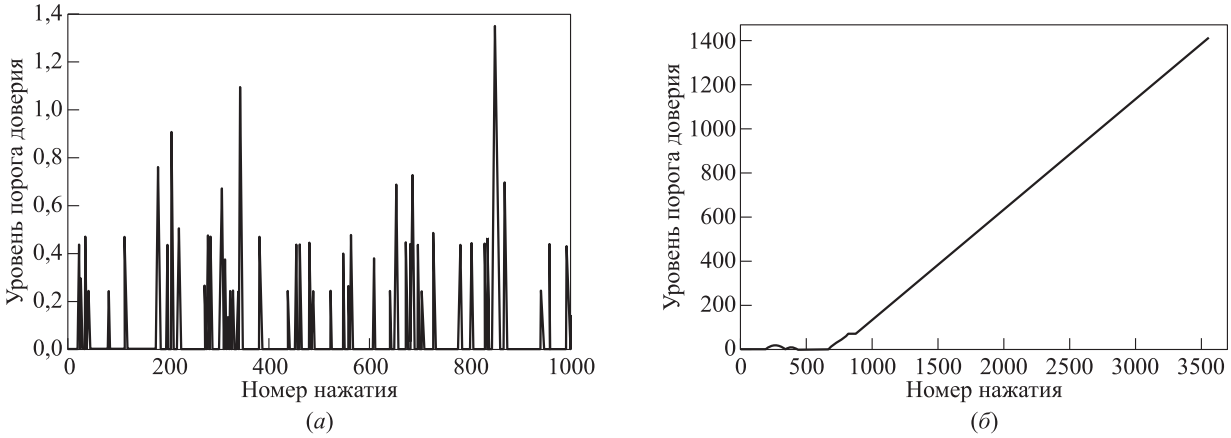



**Рис. 4** График уровня доверия для легитимного пользователя (*а*) и для атакующего набора (*б*)

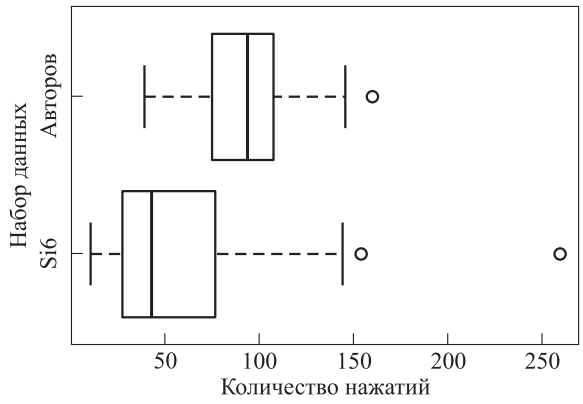



**Рис. 5** Количество нажатий, необходимое для обнаружения нарушителя

тель был определен уже после 50-го и 100-го нажатия (для Si6 и данных, собранных авторами, соответственно).

# 6 Заключение

В статье проведен обзор методов статической и динамической аутентификации с учетом динамики нажатий клавиш. Среди методов статической аутентификации рассмотрены подходы, учитывающие время удержания клавиш, порядок нажатия и отпускания кнопок; методы, принимающие решение об успешности аутентификации на основе относительной скорости печати и использовании левой и правой клавиш Shift. Кроме того, рассмотрен подход, дающий высокие результаты на коротких паролях, и оценено влияние информированности пользователя на результаты.

В рамках задачи динамической аутентификации рассматривался метод, основанный на кластеризации, метод, который использует комбинацию метрик расстояния между очередным нажатием и шаблоном пользователя, показавший хорошие результаты. Также рассматривался подходящий для задачи непрерывного контроля пользователя алгоритм, применяющий функцию наказания-поощрения.

Как альтернатива существующим методам была предложена модель представления данных, основанная на потенциальных функциях вместе с алгоритмами, на которых ее применение дает высокие результаты (алгоритмы деревьев решений C5.0 и случайный лес). Для задачи динамической аутентификации был использован метод поощрения и наказания. Для проверки эффективности сочетаний модели и перечисленных алгоритмов были проведены эксперименты на данных из набора Si6 и данных, собранных авторами в рамках этой исследовательской работы.

Проведенные эксперименты показали применимость предложенного подхода для динамической аутентификации. Таким образом, авторами статьи было признано целесообразным создание экспериментальной системы аутентификации, основанной на механизме, предложенном в ней.

# Литература

1. *Lau E., Liu X., Xiao C., Yu X.* Enhanced user authentication through keystroke biometrics. — Massachusetts Institute of Technology, 2004.
2. *Saggio G., Costantini G., Todisco M.* Cumulative and ratio time evaluations in keystroke dynamics to improve the

password security mechanism // J. Computer Information Technol., 2011. Vol. 1. No. 2. P. 4–11.

3. *Sung K. S., Cho S.* GA SVM wrapper ensemble for keystroke dynamics authentication // Conference (International) on Biometrics Proceedings. — Hong Kong: ICB, 2004. P. 654–660.
4. *Leggett J., Williams G.* Verifying identity via keystroke characteristics // Int. J. Man-Machine Studies, 1988. Vol. 28. No. 1. P. 67–76.
5. *Hocquet S., Ramel J.-Yv., Cardot H.* Fusion of methods for keystroke dynamic authentication // Automatic Identification Advanced Technologies: 4th IEEE Workshop Proceedings. — Buffalo, 2005. P. 224–229.
6. *Bergadano F., Gunetti D., Picardi C.* User authentication through keystroke dynamics // ACM Trans. Information Syst. Security (TISSEC), 2002. Vol. 5. No. 4. P. 367–397.
7. *Kittler J., Hatef M., Duin R. P. W., Matas. J.* On combining classifiers // Pattern Analysis and Machine Intelligence, IEEE Trans., 1998. Vol. 20. No. 3. P. 226–239.
8. *Bertacchini M., Benitez C., Fierens P. I.* User clustering based on keystroke dynamics // XVI Congreso Argentino de Ciencias de la Computación. – Morón, 2010. P. 832–841.
9. *Bello L., Benitez C., Bertacchini M., Pizzoni J. C., Cipriano M.* Collection and publication of a fixed text keystroke dynamics dataset // XVI Congreso Argentino de Ciencias de la Computación. – Morón, 2010. P. 822–831.
10. *Rousseeuw P. J.* Silhouettes: A graphical aid to the interpretation and validation of cluster analysis // J. Comput. Appl. Math., 1987. Vol. 20. P. 53–65.
11. *Gunetti D., Picardi C.* Keystroke analysis of free text // ACM Trans. Information Syst. Security (TISSEC), 2005. Vol. 8. No. 3. P. 312–347.
12. *Monrose F., Rubin A. D.* Keystroke dynamics as a biometric for authentication // Future Generation Computer Syst., 2000. Vol. 16. No. 4. P. 351–359.
13. *Айзерман М. А., Браверман Е. М., Розоноэр Л. И.* Метод потенциальных функций в теории обучения машин. — М.: Наука, 1970.
14. *Quinlan J. R.* C4.5: Programs for machine learning. — Morgan Kaufmann, 1993.
15. *Breiman L.* Random forests // Machine Learning, 2001. Vol. 45. No. 1. P. 5–32.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 56–61

# ПРОБЛЕМЫ СЕТЕВОГО ДОСТУПА К НАУЧНЫМ ЖУРНАЛАМ

А. В. Глушановский[1], Н. Е. Калёнов[2]

**Аннотация:** Рассматриваются проблемы организации сетевого доступа российских ученых к научным журналам и базам данных. В соответствии с мировой практикой организацию такого доступа осуществляют научные библиотеки, объединяющиеся в консорциумы для получения выгодных финансовых условий. Описывается существующая в России практика организации доступа к зарубежным научным ресурсам через посредство Российского фонда фундаментальных исследований (РФФИ) и «Национального электронно-информационного консорциума» (НЭИКОН). Приведена статистика востребованности пользователями Российской академии наук (РАН) научных журналов, предоставляемых через НЭИКОН. Предложены организационные действия для решения задачи оптимизации доступа к коммерческим сетевым научным ресурсам в условиях существующих в РАН финансовых ограничений.

**Ключевые слова:** научные журналы; информация; Интернет; удаленный доступ; библиотеки; консорциум

Анализ информационных потребностей ученых РАН показывает, что по-прежнему одним из важнейших источников научной информации для них остаются научные журналы (в первую очередь — иностранные). В настоящее время наряду с традиционной печатной формой все более широкое распространение получил доступ к научным журналам через сеть Интернет.

Технически такой доступ не представляет затруднений, что создает впечатление легкой доступности полных текстов статей. На самом деле все обстоит несколько сложнее. Большинство ведущих зарубежных издательств и научных обществ, таких как Elsevier, Springer, American Physical Society, American Chemical Society и других, представляют в свободном доступе только библиографическую информацию (описание статей) и (в лучшем случае) их рефераты. Доступ к полным текстам является платным и требует заключения соответствующего договора с издательством, причем суммы таких договоров, во-первых, весьма значительны, а во-вторых, весьма заметно варьируются в зависимости от числа пользователей в организации, количества подключаемых компьютеров и ряда других параметров.

Организация сетевого доступа к коммерческим источникам научной информации требует значительной по объему и сложности специфической работы, связанной с выбором нужных ресурсов, проведением переговоров с поставщиками, согласованием условий предоставления ресурсов, заключением контрактов и оформлением лицензионных соглашений, предоставлением IP-адресов и контролем выполнения договорных обязательств. Для научных сотрудников такая деятельность не является характерной, поэтому сложившаяся мировая практика организации сетевого доступа к научной информации состоит в том, что ею занимаются библиотеки университетов, научных центров и других научных и учебных организаций. Библиотеки в силу специфики своей деятельности лучше знакомы с издательским миром, имеют опыт взаимодействия как с издательствами, так и с пользователями информации, и работа по информационному обеспечению научных исследований является их прямой обязанностью

Подобная практика сложилась и в России, в частности в РАН. Центральные академические библиотеки (такие как Библиотека Российской академии наук (БАН) в Санкт-Петербурге, Библиотека по естественным наукам РАН (БЕН РАН) в Москве, Государственная публичная научнотехническая библиотека Сибирского отделения РАН (ГПНТБ СО РАН) в Новосибирске, Центральная научная библиотека Уральского отделения РАН (ЦНБ УрО РАН) в Екатеринбурге, Центральная научная библиотека Дальневосточного отделения РАН (ЦНБ ДвО РАН) во Владивостоке), обеспечивающие информационные потребности многих институтов РАН, тематика исследований которых в значительной мере пересекается, могут получить значительно более выгодные условия доступа к научным журналам и базам данных (БД), нежели отдельные институты, заключающие самостоятельные договора с поставщиками. Кроме того, библиотеки (и/или их объединения — консорциумы) берут на себя организационные вопросы (переговоры с издательствами, заключение и оплата догово-

[1]Библиотека по естественным наукам Российской академии наук, avglush@benran.ru
[2]Библиотека по естественным наукам Российской академии наук, nek@benran.ru

ров, оформление лицензионных соглашений, сбор IP-адресов и организацию их подключения и т. д.). Библиотеки также ведут анализ фактического использования доступа и оптимизируют подписку (в условиях жестких финансовых ограничений) для своих систем в целом. Как принято в мировой практике, для оптимизации финансовых условий доступа библиотеки объединяются в консорциумы, выступающие как единое юридическое лицо в отношениях с издающими организациями (или поставщиками ресурсов).

Обычно в консорциумы объединяются библиотеки исходя из двух положений — либо предоставить узкотематическую информацию как можно более широкому кругу пользователей (объединяются организации с близкими научными интересами) и получить скидки (в расчете на одного участника консорциума) за счет значительного числа пользователей данного ресурса, либо предоставить пользователям консорциума как можно более широкий спектр информационных ресурсов (объединяются организации с различными тематическими интересами) и получить скидки (в расчете на одного участника консорциума) за счет увеличения объема предоставляемых ресурсов.

В мире существует значительное число различных библиотечных консорциумов. Международное объединение библиотечных консорциумов (The International Coalition of Library Consortia — ICOLC) [1] объединяет более 200 библиотечных консорциумов, созданных на основе коалиций библиотек по тематическому или территориальному принципу. Консорциумы бывают разной величины и типа. Например, в Финляндии практически все университетские библиотеки, библиотеки научных учреждений и публичные библиотеки объединены в FinELib [2] — национальный консорциум, поставляющий более 70% всей электронной информации [3]. Различные типы европейских библиотечных консорциумов описаны в [4].

В России в 1990–2000-е гг. сложилась аналогичная практика организации доступа к научным журналам [5]. С 1997 г. такой доступ предоставлялся в рамках консорциума, созданного по инициативе БЕН РАН и включавшего РФФИ и 14 крупнейших научных библиотек. Финансирование консорциума осуществлял РФФИ в рамках принятой в конце 1996 г. «Программы поддержки российских научных библиотек». В рамках этой программы была создана научная электронная библиотека (НЭБ). В соответствии с принципами ее организации электронные версии журналов поступали из издательств в РФФИ и загружались на специальный сервер НЭБ и его зеркала в Казани и Новосибирске. Доступ предоставлялся всем пользователям библиотек, входящих в консорциум, а поскольку в консорциум входили все центральные академические библиотеки (БАН, БЕН РАН, ГПНТБ СО РАН, ЦНБ УрО РАН и ЦНБ ДвО РАН), любой сотрудник Академии наук мог читать основные научные журналы мира. К началу 2002 г. на серверы НЭБ было загружено около 2000 наименований (около 75 000 выпусков) журналов наиболее значимых научных издательств мира [6]. Научная электронная библиотека пользовалась большой популярностью у специалистов — за год в начале 2000-х гг. из нее выгружалось около четверти миллиона статей.

Соглашение о консорциуме НЭБ, подписанное РФФИ и ведущими библиотеками, сопровождалось рядом условий, выдвинутых издательствами и направленных, в частности, на сохранение перечня приобретаемых библиотеками печатных версий журналов (по условиям участия в консорциуме организация должна была выписать для себя в печатном виде не менее 5 журналов, не входящих в подписку консорциума [7]). Имелся (и сохраняется до сих пор во всех подобного рода консорциумах) ряд ограничений на выгрузку и распространение полученных текстов (запрещается сплошное копирование номера журнала, распространение полученных материалов за пределами организации-участника).

К сожалению, в 2004 г. НЭБ РФФИ прекратила свое существование в том виде, который предусматривался соглашениями 1996 г. Причинами этого стали несколько факторов, в частности проверка РФФИ со стороны Счетной палаты. Проверка выявила нарушения Устава РФФИ, согласно которому последний не имеет права финансировать что-либо без проведения конкурсов. Это повлекло за собой проблемы финансирования поддержки технологии функционирования НЭБ (обработка и загрузка массивов данных, поддержка серверов).

С вступлением в силу 94-го Федерального закона о закупках фактически были ликвидированы механизмы координированной работы библиотек по приобретению научных ресурсов. Из-за распада консорциума наиболее значимые научные издательства отказались передавать журналы российской стороне. В результате уже загруженные на сервер журналы НЭБ были юридически переданы ООО «Научная электронная библиотека» с условием бесплатного предоставления на ее сервере (http://www.elibrary.ru), чем в настоящее время могут пользоваться российские ученые.

Российский фонд фундаментальных исследований заключил новые договора с рядом зарубежных издательств о доступе к их журналам, но уже в режиме онлайн и только для своих грантодержателей.

С этого периода и по настоящее время в России существуют два основных централизованных кана-

ла сетевого доступа учреждений РАН к зарубежной научной информации — за счет РФФИ при посредстве Внешнеэкономического объединения «Академинторг» и за счет средств Минобрнауки при посредстве НЭИКОН. Кроме централизованных источников подписки в масштабах страны доступ к зарубежным научным журналам и БД приобретают вышеперечисленные центральные библиотеки РАН за счет средств, выделяемых Президиумом РАН и руководством ее региональных отделений, а также некоторые академические институты за счет своих средств. Однако количество ресурсов, приобретаемых академическими организациями, в десятки раз меньше количества ресурсов, приобретаемых РФФИ и НЭИКОН.

В настоящее время РФФИ финансирует своим грантодержателям (на уровне организаций, через которые осуществляется оплата средств по грантам) доступ к журналам шести издательств: Wiley (1600 журналов), The American Mathematical Society (предоставляется реферативная БД MathSciNet (MSN), включающая около двух миллионов описаний статей), American Physical Society (9 журналов), Institute of Physics (49 журналов), The Royal Society of Chemistry (6 журналов), Elsevier (Freedom Collection — около 1700 журналов). До 2011 г. предоставлялся также доступ к журналам издательства Springer, но в 2012 г. РФФИ отказался от этой подписки, мотивируя это решение нехваткой финансовых средств (одновременно было сокращено число доступных журналов The Royal Society of Chemistry с 23 до 6). С 2011 г. грантодержателям РФФИ стали доступны журналы одной из коллекций издательства Elsevier (Freedom Collection — более 1700 журналов).

За счет средств РФФИ также организован доступ пяти крупнейших академических библиотек к известной БД Web of Knowledge, которая широко используется для определения публикационной активности и уровня цитирования научных публикаций.

Следует заметить, что, лишившись в 2012 г. доступа к текущим журналам издательства Springer, пользователи РФФИ лишились и доступа к журналам предыдущих лет издания, подписка на которые была ранее оплачена. Согласно условиям контракта, при прекращении подписки для доступа к ранее оплаченным журналам каждая организация должна заплатить поставщику определенную сумму в качестве компенсации затрат на поддержку его серверов.

Как указывалось выше, каждый поставщик в зависимости от суммы контракта формулирует свои условия предоставления доступа к своим ресурсам. В частности, ограничивает число пользователей, IP-адресов или количество доступных журналов. Это обусловливает ограничения для грантодержателей РФФИ в получении доступа к сетевым ресурсам. Каждый грантодержатель в начале 2012 г. должен был выбрать из предложенного РФФИ списка от одного до четырех издательств, журналы которых ему необходимы, и сообщить о своем выборе РФФИ. Последний, в зависимости от возможностей, диктуемых контрактами, принимал окончательное решение, кому и какие ресурсы предоставить.

«Национальный электронно-информационный консорциум», включающий в свой состав несколько сот организаций науки и образования, предоставляет им в 2012 г. за счет средств Министерства образования и науки доступ к полным текстам журналов следующих издательств, представляющих интерес для РАН: American Chemical Society (ACS — 38 журналов), American Institute of Physics (AIP — 10 журналов), Annual Reviews Sciences Collection (AR — 37 журналов), Business Source Complete (BSC — около 3500 журналов), Computers & Applied Sciences Complete (CASC — около 950 журналов), Nature Publishing Group (NPG — 8 журналов), Oxford University Press (OUP — более 200 журналов), Optical Society of America (OSA — 14 журналов), Sage STM (Science, Technology & Medicine — более 100 журналов), SPIE — International Society for Optics and Photonics (6 журналов и материалы конференций), Taylor & Francis (T&F — более 1000 журналов), Georg Thieme Verlag KG (Thieme — 5 журналов) The American Association for the Advancement of Science (AAAS — журнал Science).

Для сравнения — БЕН РАН на средства, выделенные ей Президиумом РАН в рамках целевого финансирования на приобретение научной литературы в 2011 г., смогла приобрести права сетевого доступа на 2012 г. лишь к 142 наименованиям журналов, отсутствующих в списках РФФИ и НЭИКОН (по соглашениям с поставщиками доступ предоставляется не только из центрального здания БЕН РАН, но и из ее отделов, расположенных в научных учреждениях РАН).

«Национальный электронно-информационный консорциум», работая по контракту с Минобрнауки, уделяет серьезное внимание анализу использования ресурсов, предоставляемых научным организациям. Эту работу, касающуюся академических учреждений, с 2010 г. по договору с НЭИКОН проводит БЕН РАН. В ходе проводимого анализа был получен ряд интересных предварительных (работы заканчиваются в 2013 г.) результатов, которые приведены ниже.

По 14 издательствам, используемым в учреждениях РАН в 2010 г., в среднем в месяц выгружалось

**Таблица 1** Активность использования ресурсов

| Ресурс | Количество журналов | Количество выгрузок в месяц |
|---|---|---|
| ACS | 38 | 23 806 |
| AIP | 10 | 12 930 |
| NPG | 8 | 6378 |
| T&F | 1547 | 3236 |
| AAAS (Science) | 1 | 3015 |
| OSA | 14 | 2505 |
| OUP_Full | 217 | 2106 |
| Thieme | 5 | 2035 |
| SPIE | 6 | 1568 |
| Cell | 15 | 1286 |
| Annual Review | 37 | 402 |
| SAGE | 382 | 240 |
| ACM | 420 | 91 |
| BSC | 3345 | 42 |

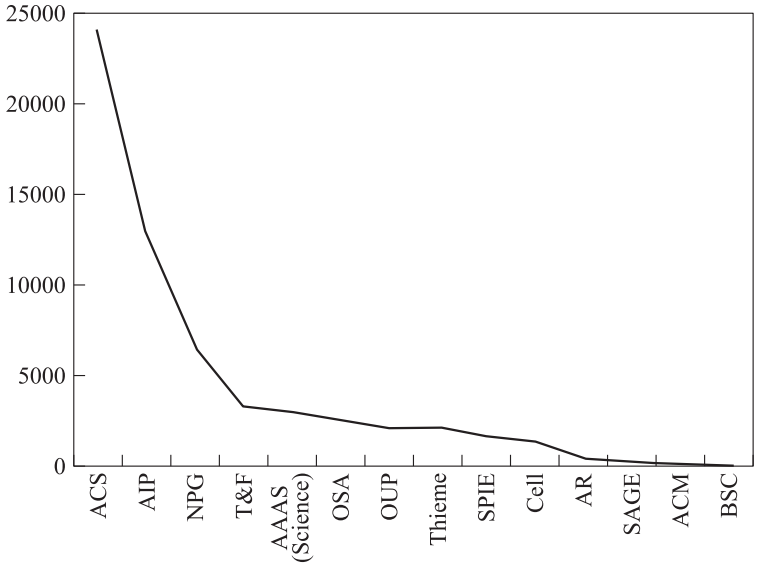
59 642 статьи. Ресурсы использовались 186 организациями РАН (журнал Science — 111 организаций, журналы AIP — 106 организаций, журналы ACS — 82 организации, журналы группы Nature (NPG — Nature Publishing Group) — 70 организаций и т. д.). В целом, в 2010 г. активность учреждений РАН, измеряемая числом выгрузок полных текстов статей в месяц, выглядит следующим образом (табл. 1).

По данным табл. 1 представлен график (см. рисунок).

Данный график имеет две точки перегиба (после NPG и после группы журналов Cell). Суммарное число выгрузок до первой точки перегиба составляет 72% от общего количества статей, выгруженных РАН, а до второй — 97%.

Наибольшим спросом у ученых РАН пользуются журналы ACS, AIP и NPG. Наименьшим спросом — журналы издательств Business Source Complete, Association for Computing Machinery (ACM), Sage и Annual Review. В средней части таблицы — пользующиеся, тем не менее, заметным спросом журналы издательств T&F, AAAS (Science), OSA, OUP, Thieme, SPIE и Cell.

До настоящего времени доступ к журналам, выписываемым через НЭИКОН, предоставлялся бесплатно. Со второй половины 2012 г. НЭИКОН планирует (по указанию Минобрнауки) взимать часть стоимости ресурсов с получателей. В этой ситуации станет актуальным вопрос, не дешевле ли будет вместо оплаты доступа к базам малоспрашиваемых журналов заказывать электронные копии отдельных статей с «постатейной» оплатой. Этот вид сервиса достаточно хорошо развит за рубежом, им пользуются как отдельные ученые, так и научные библиотеки по заказам своих пользователей. Как правило, к оглавлениям и аннотациям статей из научных журналов предоставляется свободный доступ. Функции «посредника», осуществляющего прием заказов на статьи от ученых РАН, контакты с поставщиками, оплату заказов в валюте могли бы взять на себя центральные академические библиотеки. По данным БЕН РАН, стоимость электронной копии статьи объемом до 20 страниц в библиотеках континентальной Европы составляет в среднем около 10 евро, что при годовых объемах 50–60 статей будет существенно дешевле, чем




Распределение числа выгрузок по издательствам

оплата доступа ко всем журналам издательства, не являющегося приоритетным для РАН. Очевидно, что такой подход потребует некоторого перераспределения средств и организационной перестройки библиотечных служб, но он может оказаться достаточно эффективным.

«Национальный электронно-информационный консорциум» достаточно оперативно реагирует на изменения спроса на журналы. Так, по результатам проведенного анализа с 2011 г. была прекращена подписка на журналы ACM. Что касается журналов издательств Business Source Complete, они пользуются значительным спросом у второй большой группы организаций–пользователей НЭИКОН: российских университетов. Издательство Sage, которое также оказалось в нижней части рейтинговой таблицы, в 2010 г. было представлено в НЭИКОН только журналами по гуманитарным и социальным наукам, хотя выпускает оно и другую научную литературу. В настоящее время НЭИКОН предполагает дополнить этот ресурс журналами группы Sage STM (Science, Technology & Medicine), и, возможно, результаты его востребованности академическими организациями изменятся.

Получив полные данные о спросе на журналы НЭИКОН, авторы статьи провели анализ их использования сотрудниками отделений РАН. В табл. 2 представлены показатели Отделения нанотехнологий и информационных технологий (ОНИТ).

В табл. 2 более подробно, чем в предыдущей, раскрыты журналы группы Nature, а также журналы издательств Taylor & Francis и Oxford University Press, поэтому таблица формально включает 30 ресурсов, но фактически это те же 14 ресурсов, раскрытых более подробно.

Как видно из табл. 2, наибольший интерес для сотрудников ОНИТ представляют журналы AIP и OSA. Далее со значительным отрывом следуют журналы Society of Photographic Instrumentation Engineers. Среднюю группу (30–40 обращений к полным текстам в месяц) составляют журналы American Chemical Society, AAAS (Science) и основной

**Таблица 2** Использование ресурсов ОНИТ РАН

| Ресурс | Число выгрузок в месяц |
|---|---|
| American Institute of Physics (AIP) | 413 |
| Optical Society of America (OSA) | 331 |
| Society of Photographic Instrumentation Engineers (SPIE) | 94 |
| American Chemical Society (ACS) | 42 |
| AAAS (Science) | 36 |
| Nature | 31 |
| Sage | 18 |
| Nature Physics | 17 |
| Nature Nanotechnology | 13 |
| Association for Computing Machinery (ACM) | 10 |
| Nature Photonics | 8 |
| Nature Materials | 7 |
| Nature Chemistry | 0,25 |
| Nature Methods | 0,17 |
| Business Source Complete | 0 |
| Cell | 0 |
| Nature Biotechnology | 0 |
| Oxford University Press. Mathematics & Computing | 0 |
| Oxford University Press BioMed | 0 |
| Oxford University Press Life | 0 |
| Oxford University Press Med | 0 |
| Oxford University Press STM | 0 |
| Taylor & Francis Bio | 0 |
| Taylor & Francis Chem | 0 |
| Taylor & Francis Earth | 0 |
| Taylor & Francis. Natural Sciences | 0 |
| Taylor & Francis Med | 0 |
| Taylor & Francis. Other | 0 |
| Taylor & Francis. Physics & Mathematics | 0 |
| Taylor & Francis. Technique | 0 |

журнал группы Nature. Значительно меньшим спросом пользуются остальные журналы группы Nature, журналы же остальных издательств не представляют для ОНИТ никакого интереса.

Российский фонд фундаментальных исследований, в отличие от НЭИКОН, не предоставляет пользователям статистики использования представляемых Фондом ресурсов (хотя БЕН РАН обращалась по этому поводу к руководству РФФИ и получила принципиальное согласие, но данные пока не получила), поэтому аналогичный анализ по этим ресурсам пока невозможен. Однако, по наблюдениям авторов статьи, журналы, предлагаемые через РФФИ, пользовались также весьма заметным спросом. Так, статьи из журналов издательства Springer в 2011 г. только в БЕН РАН выгружались в среднем 158 раз в месяц, в ГПНТБ СО РАН — 489 раз; статьи издательства Elsevier в 2011 г. выгружались в БЕН РАН в среднем 1536 раз в месяц.

Таким образом, в настоящее время в России создана действующая система доступа к полным текстам нескольких тысяч зарубежных научных журналов. Эта система охватывает подавляющее большинство научных организаций РАН и пользуется значительной популярностью. Однако она не охватывает (в первую очередь в силу недостаточного финансирования) весь необходимый объем научной информации, требуемый для эффективного функционирования научных институтов и центров РАН. В существующих условиях, к сожалению, не представляется возможным обеспечить доступ с каждого рабочего места сотрудника РАН ко всем необходимым ему журналам. Оптимизировать систему возможно за счет серьезного анализа фактического спроса, создания ранжированного списка наиболее востребованных журналов, выявления необходимых издательств и централизованного (на базе существующих или вновь создаваемых консорциумов) заключения договоров с этими издательствами.

Другим фактором оптимизации системы является сокращение числа пользователей (IP-адресов) каждого научного института (научного центра), получающих доступ к тому или иному ресурсу, что позволит снизить стоимость договоров с поставщиками. В этом плане представляется целесообразным подход, реализованный в БЕН РАН, которая при заключении договоров оговаривает права доступа к ресурсам не только из здания Центральной библиотеки, но и из ее отделов (библиотек) в научно-исследовательских учреждениях (НИУ) РАН. При этом поставщикам официально сообщаются IP-адреса библиотечных компьютеров, с которых сотрудники институтов могут читать журналы. Такая схема работает уже несколько лет и позволяет без существенных затрат обеспечивать важнейшей информацией (хотя и не с каждого компьютера института) сотрудников более 40 институтов и научных центров Москвы и Московского региона. Существенным ограничением этой схемы является требование поставщиков, чтобы отделы БЕН в НИУ РАН имели компьютеры с выделенными IP-адресами, поэтому библиотекам, работающим через прокси-серверы институтов, доступ к ресурсам предоставлен быть не может.

Необходимо отметить, что в России и, в частности, в РАН доля финансирования, выделяемая на информационное обеспечение науки, существенно меньше принятой в развитых и развивающихся странах. Согласно мировой практике эта доля составляет от 8% до 12% от ассигнований на научные исследования. У нас она не достигает и 1%.

В существующих условиях, когда библиотекам катастрофически не хватает централизованно выделяемых РАН средств на приобретение информационных ресурсов, необходимо и впредь развивать идеи создания академических и межведомственных консорциумов по доступу к научной информации, интеграции финансов, выделяемых библиотекам, и собственных финансов НИУ, перехода на новые системы информационного обслуживания пользователей.

## Литература


1. The International Coalition of Library Consortia (ICOLC). http://www.library.yale.edu/consortia.
2. FinELib, the National Electronic Library. The National Library of Finland. http://www.nationallibrary.fi/libraries/finelib/finelibconsortium.html.
3. *Hökli E.* Libraries in Finland establish consortia // Liber Quarterly: The J. European Research Libraries, 2001. Vol. 11. No. 1. P. 53–59.
4. *Hormia-Poutanen K., Xenidou-Dervou C., Kupryte R., Stange K., Kuznetsov A., Woodward H.* Consortia in Europe: Describing the various solutions through four country examples // Library Trends, 2006. Vol. 54. No. 3. https://dspace.lib.cranfield.ac.uk/handle/1826/1014.
5. *Литвинова Н. Н.* Электронные документы: отбор, использование и хранение // Библиотека, 2005. № 6. С. 6–9.
6. *Никаньшин Д. П., Туриянский И. Е., Астафьев М. Н.* О развитии зеркального сервера научной электронной библиотеки РФФИ // Исследования по информатике, 2003. Вып. 5. С. 133–142.
7. *Хельферих П., Красикова О. Л.* Научная информация для российских библиотек // Библиотеки и ассоциации в меняющемся мире: новые технологии и новые формы сотрудничества: Мат-лы 7-й Междунар. конф. — Судак, Крым, Украина, 2000. — Т. 2. С. 127–128.

# МОДЕЛИРОВАНИЕ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ СИНЕРГЕТИЧЕСКИМ ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ

И. А. Кириков[1], А. В. Колесников[2], С. В. Листопад[3]

**Аннотация:** Рассматривается подход к моделированию коллективных эффектов систем поддержки принятия решений в рамках синергетической парадигмы искусственного интеллекта. Приведена модель и функциональная структура гибридной интеллектуальной многоагентной системы (ГиИМАС) для моделирования систем поддержки принятия решений (СППР). Представлены результаты вычислительных экспериментов, демонстрирующие положительное влияние эффекта самоорганизации на качество коллективных решений.

**Ключевые слова:** компьютерная система поддержки принятия решений; гибридная интеллектуальная многоагентная система с самоорганизацией

## 1 Введение

Люди ежедневно сталкиваются с принятием решений. Одни решения даются легко, часто без обдумывания, в некоторых случаях обращаются к коллегам, справочной литературе и другим источникам информации. Это и называется поддержкой принятия решений [1]. В наиболее сложных случаях нужно учитывать противоречивые требования, оценивая множество альтернатив по нескольким критериям. Тогда создается СППР — коллектив экспертов под управлением лица, принимающего решения (ЛПР). Концептуальная модель СППР показана на рис. 1 [2].

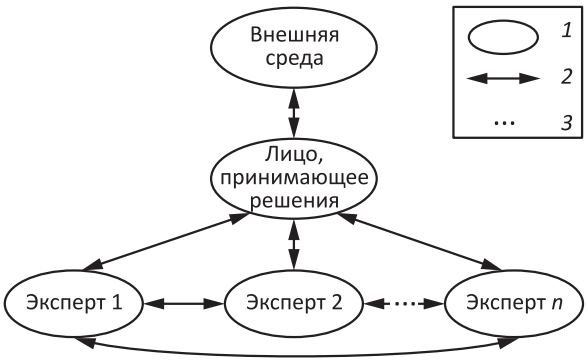



**Рис. 1** Концептуальная модель СППР: *1* — участники СППР; *2* — отношения между участниками СППР; *3* — возможность расширения количества экспертов СППР и установления отношений между ними и другими участниками СППР

Согласно данному определению СППР, процесс принятия решений в ней — частный случай коллективного принятия решений в малых группах, изучаемых со второй четверти XX в. социальной психологией и социологией, где получены результаты по увеличению эффективности работы индивидов в группе, а также выявлению возникающих при этом коллективных эффектов [3, 4].

В то время как люди уже давно решают сложные задачи только коллективно, в СППР создаваемые интеллектуальные информационные системы все еще не релевантны этим целям [5]. В этой связи актуально научить ЭВМ работать в условиях сложных задач не хуже коллектива специалистов [6], для чего важны исследования коллективных эффектов, построение их моделей и создание компьютерных СППР (КСППР).

## 2 Коллективные эффекты в системах поддержки принятия решений

Коллективные эффекты — механизмы функционирования коллектива людей, за счет которых осуществляются коллективные процессы и достигаются коллективные состояния; средства, интегрирующие индивидуальные действия в коллективной работе и общении [7] (ниже названы

[1]Калининградский филиал Института проблем информатики Российской академии наук, baltbipiran@mail.ru
[2]Калининградский филиал Института проблем информатики Российской академии наук, avkolesnikov@yandex.ru
[3]Калининградский филиал Института проблем информатики Российской академии наук, ser-list-post@yandex.ru

макроуровневыми процессами). В результате анализа литературы по социальной психологии [7–10] и синергетическому искусственному интеллекту [5, 11, 12] выявлено 12 основных коллективных эффектов (табл. 1).

Как показывает анализ табл. 1, все эффекты в той или иной степени сказываются на качестве принимаемых в СППР решений. Особую и основополагающую роль среди них играет самоорганизация — предпосылка возникновения других коллективных эффектов, положительно влияющих на результат работы СППР, например синергии [5].

Как показал анализ подходов к моделированию СППР с использованием методов синергетического искусственного интеллекта [5], среди которых гибридные системы [13], интегрированные экспертные системы [14], гибридные интеллектуальные системы (ГиИС) [2], многоагентные системы

**Таблица 1** Коллективные эффекты в СППР

| Эффект | Краткое описание | Позитивное влияние | Негативное влияние |
|---|---|---|---|
| 1. Бумеранг | При недоверии к информации возникает мнение, обратное содержащемуся в ней | Информация не воспринимается либо считается заведомо ложной | Достоверная информация от недостоверного источника может быть расценена как ложная |
| 2. Волна | Распространение идей в СППР, отвечающих интересам ее членов | Коллективная доработка идей | Длительная работа экспертов над бесперспективными идеями |
| 3. Групповой эгоизм | Цели коллектива важнее целей ее члена и целей общества | | Эффективная деятельность коллектива может вредить обществу |
| 4. Групсинк и конформизм | Общее мнение — истина, мнение отдельного человека — ничто | | Препятствует возникновению новых подходов к решению проблемы |
| 5. Мода (подражание) | Добровольное принятие установившейся в коллективе точки зрения на проблему | Основа самообучения членов коллектива; способствует адаптации людей друг к другу | Снижает вероятность появления оригинальных взглядов и подходов к решению проблемы |
| 6. Ореол | В условиях дефицита времени объекты могут восприниматься на основе стереотипов | Быстрое формирование мнения (модели) о некоторой сущности | Формируемые модели неточны и могут приводить к ошибочным решениям |
| 7. Принадлежность к группе | Члены коллектива реагируют на других людей с позиций коллектива, а не с позиций личности | Способствует сближению моделей внешнего мира у членов группы и упрощает взаимодействие между ними | Гиперболизация эффекта ведет коллектив к переоценке возможностей, отрыву от других коллективов |
| 8. Пульсар | Изменение активности коллектива в зависимости от стимулов | Эффективное решение срочных задач (рост скорости при том же качестве), снижение активности в другое время | На отдельных этапах решения задачи возникает перенапряжение членов коллектива |
| 9. Рингельмана | С ростом численности группы снижается индивидуальный вклад в общую работу | Снижается нагрузка на отдельных участников СППР | Снижается мотивация экспертов к эффективной совместной работе |
| 10. Самоорганизация | Отношения между экспертами динамичны и изменяются в процессе работы | Адаптация к внешней среде: каждый раз вырабатывать новый метод, релевантный задаче. Появление оригинальных подходов к решению задачи и синергии | Затрудняет анализ работы и внешнее управление коллективом |
| 11. Синергия | Получение общего результата, который не могут получить эксперты индивидуально | Получение эмерджентного качественно лучшего результата | Возможен эффект отрицательной синергии (дисергии) |
| 12. Социальная фасилитация | Усиление доминантных реакций в присутствии других людей | Ускоряет решение простых задач, на которые индивид знает ответ | В сложных задачах повышает вероятность ошибочного ответа |

(МАС) [11], все они позволяют моделировать те или иные понятия, используемые при описании СППР. Тем не менее только МАС релевантны моделированию коллективных эффектов СППР, однако в МАС не оговаривается, какие интеллектуальные технологии должны использовать агенты МАС. В случае гомогенных агентов МАС будет неспособна решать сложные неоднородные задачи, так как многие аспекты сложной задачи выпадут из «поля зрения» всех ее агентов. В связи с этим в данной работе в качестве модели СППР исследуется новый класс интеллектуальных систем — ГиИМАС. Они объединяют преимущества МАС и ГиИС, с одной стороны, имитируя коллективные эффекты в СППР, а с другой — интегрируя разнородные технологии моделирования интеллектуальной деятельности человека.

## 3 Модель самоорганизации в системах поддержки принятия решений

Самоорганизация в СППР рассматривается как изменение ЛПР отношений между экспертами на основе анализа их целей:

$$\begin{aligned}\mathrm{so}^{\mathrm{goa}} = r_2^{\mathrm{res\,act}}(\mathrm{dss}, \mathrm{ACT}^{\mathrm{sen}})\circ \\ \circ\, r_1^{\mathrm{act\,res}}(\mathrm{ACT}^{\mathrm{sen}}, \mathrm{env}) \circ R_1^{\mathrm{res\,res}}(\widetilde{\mathrm{DSS}}, \widetilde{\mathrm{DSS}})\circ \\ \circ\, r_3^{\mathrm{res\,res}}(\mathrm{dss}, \mathrm{prt}^{\mathrm{dm}}) \circ r_2^{\mathrm{res\,act}}(\mathrm{prt}^{\mathrm{dm}}, \mathrm{act}_{\mathrm{ia}})\circ \\ \circ\, r_1^{\mathrm{act\,res}}(\mathrm{act}_{\mathrm{ia}}, \widetilde{\mathrm{dss}}_{\mathrm{cur}})\circ \\ \circ\, r_2^{\mathrm{res\,act}}(\mathrm{prt}^{\mathrm{dm}}, \mathrm{act}_{\mathrm{ac}}) \circ r_1^{\mathrm{act\,res}}(\mathrm{act}_{\mathrm{ac}}, \widetilde{\mathrm{DSS}})\circ \\ \circ\, r_2^{\mathrm{act\,res}}(\mathrm{act}_{\mathrm{ac}}, \widetilde{\mathrm{dss}}_{\mathrm{des}})\,,\end{aligned}$$

где $\mathrm{so}^{\mathrm{goa}}$ — концептуальная модель самоорганизации СППР на основе анализа целей экспертов; dss — концептуальная модель СППР; $\mathrm{ACT}^{\mathrm{sen}}$ — множество действий по восприятию внешней среды; env — концептуальная модель внешней среды; $\widetilde{\mathrm{DSS}}$ — множество возможных в данной СППР ситуаций коллективного решения; $\mathrm{prt}^{\mathrm{dm}}$ — ЛПР; $\mathrm{act}_{\mathrm{ia}}$ — действие ЛПР «идентификация текущей ситуации коллективного решения»; $\widetilde{\mathrm{dss}}_{\mathrm{cur}}$ — текущая ситуация коллективного решения; $\mathrm{act}_{\mathrm{ac}}$ — действие ЛПР «выбор желаемой ситуации коллективного решения из множества возможных»; $\widetilde{\mathrm{dss}}_{\mathrm{des}}$ — желаемая ЛПР с точки зрения параметров задачи и его знаний об эффективности той или ситуации из DSS; $r_2^{\mathrm{res\,act}}$ — отношение «выполнять» для субъекта и выполняемого им действия; $r_1^{\mathrm{res\,act}}$ — отношение «иметь объектом» для действия и его ресурса; $R_1^{\mathrm{res\,res}}$ — множество отношений «следовать за» класса «ресурс–ресурс»; $r_3^{\mathrm{res\,res}}$ — отношение «включать»; $r_2^{\mathrm{act\,res}}$ — отношение «иметь результатом» для действия и его результата.

Процесс самоорганизации в СППР моделируется с использованием ГиИМАС, определяемой следующим образом:

$$\mathrm{himas} = \langle \mathrm{AG}^*, \mathrm{env}, \mathrm{INT}^*, \mathrm{ORG}, \{\mathrm{so}^{\mathrm{goa}}\}\rangle\,; \quad (1)$$

$$\mathrm{AG}^* = \{\mathrm{ag}_1, \ldots, \mathrm{ag}_n, \mathrm{ag}^{\mathrm{dm}}\}\,; \quad (2)$$

$$\mathrm{INT}^* = \{\mathrm{prot}, \mathrm{lang}, \mathrm{ont}, \mathrm{rcl}\}\,; \quad (3)$$

$$\left.\begin{aligned}\mathrm{ORG} = \mathrm{ORG}_{\mathrm{coop}} \cup \mathrm{ORG}_{\mathrm{neut}} \cup \mathrm{ORG}_{\mathrm{comp}}\,; \\ \mathrm{ORG}_{\mathrm{coop}} \cap \mathrm{ORG}_{\mathrm{neut}} = \varnothing\,; \\ \mathrm{ORG}_{\mathrm{coop}} \cap \mathrm{ORG}_{\mathrm{comp}} = \varnothing\,; \\ \mathrm{ORG}_{\mathrm{comp}} \cap \mathrm{ORG}_{\mathrm{neut}} = \varnothing\,;\end{aligned}\right\} \quad (4)$$

$$\begin{aligned}\mathrm{act}_{\mathrm{himas}} = \\ = \left(\bigcup_{\mathrm{ag}\in\mathrm{AG}^*} \mathrm{act}_{\mathrm{ag}}\right) \cup \mathrm{act}_{\mathrm{ia}} \cup \mathrm{act}_{\mathrm{ac}} \cup \mathrm{act}_{\mathrm{col}}\,; \quad (5)\end{aligned}$$

$$\mathrm{act}_{\mathrm{ag}} = (\mathrm{MET}_{\mathrm{ag}}, \mathrm{IT}_{\mathrm{ag}})\,, \quad \mathrm{ag} \in \mathrm{AG}^*\,, \quad \left|\bigcup_{\mathrm{ag}\in\mathrm{AG}^*} \mathrm{IT}_{\mathrm{ag}}\right| \geq 2\,; \quad (6)$$

$$\mathrm{ag} = \mathrm{ag} \vee \mathrm{himas}\,, \quad (7)$$

где $\mathrm{AG}^*$ — множество агентов ag (моделей экспертов), включающее агента, принимающего решения (АПР) — $\mathrm{ag}^{\mathrm{dm}}$, $n$ — число агентов-экспертов; env — концептуальная модель внешней среды ГиИМАС; $\mathrm{INT}^*$ — элементы структурирования взаимодействий агентов: prot — протокол взаимодействия; lang — язык передачи сообщений; ont — модель предметной области; rcl — классификатор отношений агентов (классифицирует отношения между агентами в зависимости от их целей на классы нейтралитета, конкуренции и сотрудничества); ORG — множество архитектур ГиИМАС ($\mathrm{ORG}_{\mathrm{coop}}$ — с сотрудничающими; $\mathrm{ORG}_{\mathrm{neut}}$ — с нейтральными и $\mathrm{ORG}_{\mathrm{comp}}$ — с конкурирующими агентами); $\{\mathrm{so}^{\mathrm{goa}}\}$ — множество концептуальных моделей макроуровневых процессов в ГиИМАС: $\mathrm{so}^{\mathrm{goa}}$ — модель эффекта самоорганизации на основе анализа целей; $\mathrm{act}_{\mathrm{himas}}$ — функция ГиИМАС в целом; $\mathrm{act}_{\mathrm{ag}}$ — функция агента из множества $\mathrm{AG}^*$; $\mathrm{act}_{\mathrm{ia}}$ — функция «анализ взаимодействий» АПР $\mathrm{ag}^{\mathrm{dm}}$ (модель действия ЛПР «идентификация текущей ситуации коллективного решения» $\mathrm{act}_{\mathrm{ia}}$); $\mathrm{act}_{\mathrm{ac}}$ — функция «выбор архитектуры» АПР $\mathrm{ag}^{\mathrm{dm}}$ (модель действия ЛПР «выбор желаемой ситуации коллективного решения из множества возможных»

$\mathrm{act_{ac}}$); $\mathrm{act_{col}}$ — коллективная функция ГиИМАС на межагентных отношениях $R^{\mathrm{res\,res}}$ (см. (8)) и определяемая текущей архитектурой org, конструируемая динамически в процессе функционирования системы; $\mathrm{met_{ag}}$ — метод решения задачи; $\mathrm{it_{ag}}$ — интеллектуальная технология, в рамках которой реализован метод $\mathrm{met_{ag}}$.

Для реализации функции АПР «анализ взаимодействий» $\mathrm{ag^{dm}}$ вводится формализованное понятие нечеткой цели агента pr — нечеткое множество с функцией принадлежности $\mu(\mathrm{st})$, заданное на множестве состояний ST объекта управления (ОУ). Состояние $\mathrm{st} \in \mathrm{ST}$ ОУ описывается набором его свойств $\mathrm{PR} = \{\mathrm{pr}_1, \ldots, \mathrm{pr}_{N_{\mathrm{pr}}}\}$, т. е. $\mu(\mathrm{st}) = \mu(\mathrm{pr}_1, \ldots, \mathrm{pr}_{N_{\mathrm{pr}}})$. Значение нечеткой цели определяется подстановкой в $\mu(\mathrm{pr}_1, \ldots, \mathrm{pr}_{N_{\mathrm{pr}}})$ значений свойств ОУ для данного состояния из множества $\mathrm{VAL} = \{\mathrm{val}_1, \ldots, \mathrm{val}_{N_{\mathrm{val}}}\}$, т. е. $\mu(\mathrm{val}_1, \ldots, \mathrm{val}_{N_{\mathrm{val}}})$. Вводится мера сходства нечетких целей $A$ и $B$ для одномерного случая [5]:

$$s(A,B) = \\ = \frac{1}{2}\left(\int\limits_{\mathrm{val_{min}}}^{\mathrm{val_{max}}} \mu_{A\cap B}(\mathrm{pr})\,d\,\mathrm{pr}\right)\left(\left(\int\limits_{\mathrm{val_{min}}}^{\mathrm{val_{max}}} \mu_{A}(\mathrm{pr})\,d\,\mathrm{pr}\right)^{-1} + \left(\int\limits_{\mathrm{val_{min}}}^{\mathrm{val_{max}}} \mu_{B}(\mathrm{pr})\,d\,\mathrm{pr}\right)^{-1}\right).$$

На основе значения меры сходства нечетких целей отношения между агентами могут быть классифицированы на отношения конкуренции, нейтралитета и сотрудничества, т. е. определена согласованность взаимодействия агентов. Представим класс отношения по согласованности взаимодействия нечеткими множествами на универсуме значений меры сходства целей $s$ (множестве действительных чисел в интервале [0; 1]):

$$\mu_{\text{конк}}(s) = (1 + (3s)^8)^{-1};$$
$$\mu_{\text{нейтр}}(s) = (1 + (6(s - 0{,}5))^8)^{-1};$$
$$\mu_{\text{сотр}}(s) = (1 + (3(s-1))^8)^{-1}.$$

Класс отношения между агентами ГиИМАС по согласованности их взаимодействия представим лингвистической переменной

$$\mathrm{cl} = \langle \beta, T, U, G, M \rangle,$$

где $\beta$ = «класс отношений» — наименование лингвистической переменной; $T$ = {«конкуренция»; «нейтралитет»; «сотрудничество»} — терм-множество ее значений, каждое из которых — название нечеткой переменной; $U = [0;1]$ — универсум нечетких переменных; $G = \varnothing$ — процедура образования из элементов множества $T$ новых термов; $M = \{\mu_{\text{конк}}(s), \mu_{\text{нейтр}}(s), \mu_{\text{сотр}}(s)\}$ — процедура, ставящая в соответствие каждому терму множества $T$ осмысленное содержание путем формирования соответствующего нечеткого множества.

Когда для каждой пары агентов ГиИМАС определено значение cl и составлена матрица **CL** (матрица классов отношений), она анализируется, чтобы идентифицировать текущую архитектуру ГиИМАС: с сотрудничающими, нейтральными или конкурирующими агентами [5]. В зависимости от параметров задачи АПР стремится установить одну из них, чтобы повысить эффективность работы ГиИМАС.

В результате имитации процессов самоорганизации определяется архитектура ГиИМАС:

$$\mathrm{org} = R^{\mathrm{res\,res}}(\mathrm{AG}^*, \mathrm{env}) \circ R^{\mathrm{res\,res}}(\mathrm{AG}^*, \mathrm{AG}^*), \quad (8)$$

где $R^{\mathrm{res\,res}}$ — множество отношений «ресурс–ресурс».

Учитывая, что функция $\mathrm{act_{ag}}$ агента $\mathrm{ag} \in \mathrm{AG}^*$ выполняется множеством методов $\mathrm{MET_{ag}}$, концептуальная модель ГиИМАС как метода решения сложной задачи представляется выражением:

$$\mathrm{MET_{himas}} = R^{\mathrm{met\,met}}(\mathrm{MET_{ag_i}}, \mathrm{MET_{ag_j}}), \\ \mathrm{ag}_i, \mathrm{ag}_j \in \mathrm{AG}^*, \ \mathrm{ag}_i \neq \mathrm{ag}_j,$$

где метод $\mathrm{MET_{himas}}$, вырабатываемый ГиИМАС при решении сложной задачи, — взаимосвязанная совокупность реализуемых агентами методов $\mathrm{MET_{ag}}$. При решении каждой задачи АПР анализирует взаимодействия $\mathrm{act_{ia}}$ между агентами, выбирает архитектуру $\mathrm{act_{ac}}$, определяя интенсивность и согласованность отношений $R^{\mathrm{met\,met}}$ между моделями знаний агентов, что рассматривается как выработка нового метода, релевантного ситуации решения сложной задачи.

Для реализации функции АПР «выбор архитектуры» $\mathrm{ag^{dm}}$ необходима база знаний о релевантности архитектуры ГиИМАС ситуации решения задачи:

$$\mathrm{act_{ac}} = r_1^{\mathrm{act\,mod}}(\mathrm{act_{ac}}, \mathrm{mod_{ac}}) \circ r_1^{\mathrm{act\,alg}}(\mathrm{act_{ac}}, \mathrm{alg_{ac}}),$$

где $\mathrm{mod_{ac}}$ — модель нечеткого вывода; $\mathrm{alg_{ac}}$ — алгоритм функции «выбор архитектуры»; $r_1^{\mathrm{act\,mod}}$ — отношение между действием и его моделью; $r_1^{\mathrm{act\,alg}}$ — отношение действия и его алгоритма. При разработке модели нечеткого вывода использован нечеткий вывод Мамдани [1] с самообучением, в результате которого определяются четкие оценки степеней уверенности АПР в выборе архитектуры

ГиИМАС, релевантной решаемой задаче. Используемый алгоритм самообучения модели нечеткого вывода Мамдани основан на методе обратного распространения ошибки [1], применяющемся при обучении нейронных сетей.

В итоге суть метода моделирования самоорганизации с использованием ГиИМАС состоит в выполнении логически упорядоченной последовательности нижеперечисленных действий:

(1) анализ взаимодействий агентов;

(2) выбор архитектуры:

(2.1) получение исходных данных;

(2.2) вычисление по алгоритму нечеткого вывода Мамдани значений степеней уверенности в выборе архитектуры;

(2.3) выбор архитектуры с вероятностью, пропорциональной степеням уверенности из (2.2);

(2.4) имитационный процесс решения задачи на архитектуре из (2.3);

(2.5) вычисление значения абсолютной ошибки нечеткого вывода в (2.2);

(2.6) корректировка функций принадлежности нечетких переменных по ошибке.

## 4 Гибридная интеллектуальная многоагентная система для моделирования самоорганизации в системах поддержки принятия решений

Для компьютерной реализации модели (1)–(7) разработана универсальная функциональная структура ГиИМАС (рис. 2). Она может быть использована для широкого круга задач, поскольку

(1) использована общая многоагентная модель действительности;

(2) перечень агентов-решателей охватывает пять классов методов из шести, используемых в ГиИС [2];

(3) порядок взаимодействия агентов определяется моделью предметной области и конкретными реализуемыми агентами алгоритмами, не специфицированными данной архитектурой, которые реализуются разработчиком автоматизированной системы для решения поставленной задачи.

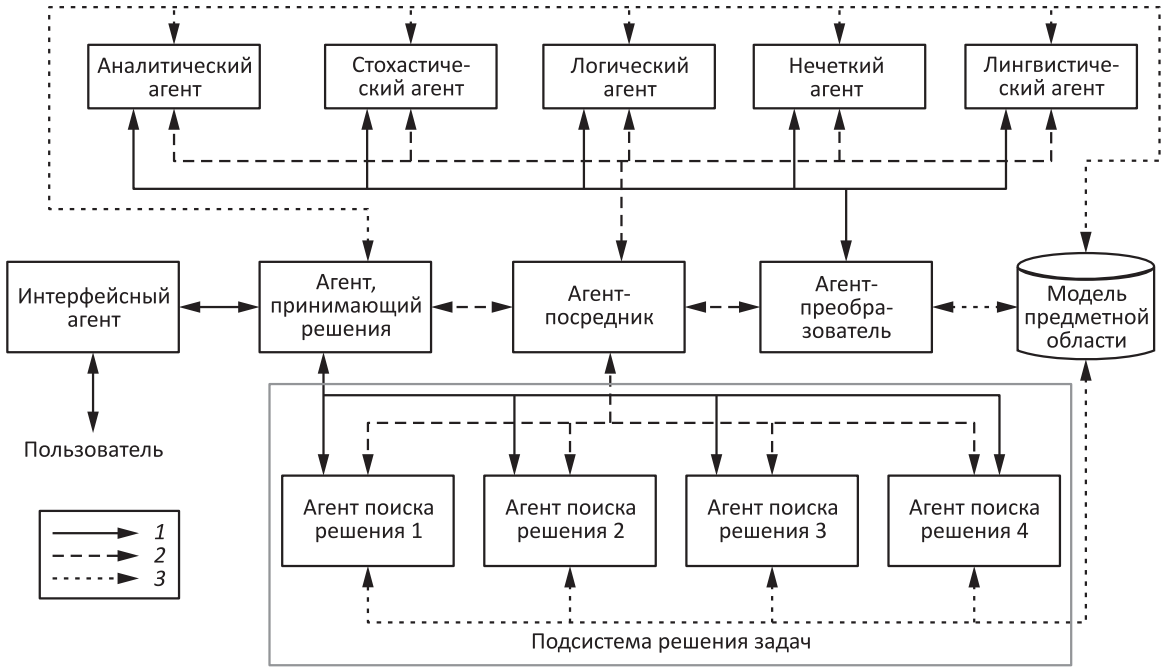



**Рис. 2** Универсальная функциональная структура ГиИМАС с самоорганизацией: *1* — взаимоотношения агентов (запросы информации, передача результатов их решения); *2* — взаимоотношения агентов (запросы помощи в решении подзадач); *3* — взаимодействие (получение сведений из модели, обновление модели) агентов с моделью предметной области

Рассмотрим назначение ее агентов:

- интерфейсный агент запрашивает входные данные и выдает результат;
- АПР рассылает агентам поиска решения (поисковикам) условия задачи, определяет порядок их взаимодействия с помощью функций «анализ взаимодействия» и «выбор архитектуры». Когда последние решили задачу, он выбирает альтернативу и передает интерфейсному агенту или запускает новую итерацию решения задачи, рассылая решение остальным агентам поиска;
- агенты поиска решения имеют знания о предметной области и используют муравьиный алгоритм [5] для решения подзадач;
- агент-посредник отслеживает имена, модели и возможности зарегистрированных агентов интеллектуальных технологий (решателей). Агенты обращаются к нему, чтобы узнать, какой из решателей может помочь в решении поставленной перед ними подзадачи;
- решатели из верхней части рис. 2 вместе с агентом-преобразователем реализуют гибридную составляющую ГиИМАС, комбинируя разнородные знания, и предоставляют «услуги» агентам с использованием следующих моделей и алгоритмов: алгебраических уравнений для описания причинно-следственных связей концептов предметной области; метода Монте-Карло; продукционной экспертной системы с рассуждениями в прямом направлении; алгоритма нечеткого вывода Мамдани;
- модель предметной области — семантическая сеть, основа взаимодействия агентов, построена по концептуальной модели решаемой задачи. Агенты интерпретируют смысл получаемых сообщений на этой модели.

## 5 Результаты экспериментов

Для оценки влияния эффекта самоорганизации на качество решений ГиИМАС проведены серии экспериментов, в которых требовалось решить сложную транспортно-логистическую задачу, т. е. найти для нескольких транспортных средств совокупность маршрутов, оптимальную по четырем критериям:

(1) суммарная стоимость маршрута;

(2) общая длительность поездок для всех транспортных средств;

(3) вероятность опоздания хотя бы к одному клиенту;

(4) надежность (мерой надежности будем считать математическое ожидание увеличения стоимости совокупности маршрутов) [5].

Учитывались такие стохастические факторы, как вероятность возникновения дорожных пробок и вероятность опоздания к клиенту, потери от боя груза и др.

Исходные данные:

(1) запросы клиентов на доставку грузов: наименование, количество товара, временной интервал его доставки;

(2) сведения о дорогах к клиентам: протяженность, загруженность, качество;

(3) паспортные данные транспортных средств: расход горюче-смазочных материалов, грузоподъемность и т. п.;

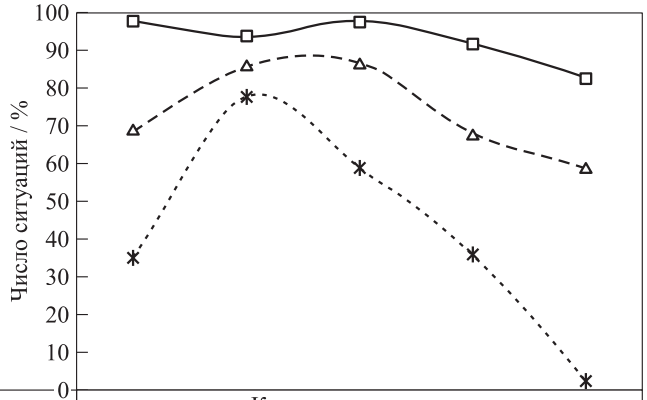
(4) сведения о графиках работы и заработной плате персонала: водителей и грузчиков;

(5) информация о грузе: вес, габариты, хрупкость и т. п.

Выходные данные: совокупность маршрутов доставки грузов (по одному на транспортное средство) и ее параметры: стоимость, длительность, надежность и вероятность опоздания, сводный критерий качества маршрута (среднее значение нечеткой цели АПР). Для тестирования разработаны пять задач, количественные параметры которых приведены в табл. 2.

**Таблица 2** Количественные параметры тестируемых задач

| Задача | Количество клиентов | Количество дорог | Количество водителей | Количество грузчиков | Количество транспортных средств |
|---|---|---|---|---|---|
| 3_10 | 10 | 75 | 3 | 3 | 3 |
| 3_15 | 15 | 240 | 5 | 5 | 5 |
| 3_20 | 20 | 420 | 5 | 5 | 5 |
| 3_25 | 25 | 650 | 9 | 9 | 9 |
| 3_30 | 30 | 377 | 6 | 6 | 6 |




| Архитектура ГиИМАС | Количество клиентов | | | | |
|---|---|---|---|---|---|
| | 10 | 15 | 20 | 25 | 30 |
| с конкуренцией | 35 | 78 | 59 | 36 | 2 |
| с нейтралитетом | 98 | 94 | 98 | 92 | 83 |
| с сотрудничеством | 69 | 86 | 87 | 68 | 59 |

**Рис. 3** Зависимость числа ситуаций, когда коллективное решение лучше любого индивидуального, от числа клиентов в задаче

Исследовались три архитектуры ГиИМАС: с нейтральными, сотрудничающими и конкурирующими агентами. В ГиИМАС с нейтральными агентами каждый из четырех агентов поиска решения минимизирует значение «своего» критерия оценки решения. В ГиИМАС с сотрудничающими агентами все четыре агента-поисковика минимизируют все четыре критерия оценки решения (аналогично АПР). В ГиИМАС с конкурирующими агентами один агент минимизирует стоимость и максимизирует длительность, второй — максимизирует стоимость и минимизирует длительность, третий — минимизирует вероятность опоздания и максимизирует надежность, а четвертый — максимизирует вероятность опоздания и минимизирует надежность.

Качество тестовых решений оценивалось по объективным показателям и субъективно экспертами. Для пяти задач и каждой архитектуры ГиИМАС проведено по 100 вычислительных экспериментов. По всем задачам и архитектурам ГиИМАС, а также для архитектуры ГиИМАС без взаимодействия (агенты-поисковики не обменивались индивидуальными решениями) построены графические зависимости среднего значения цели агента, принимающего решения, средних значений стоимости, длительности, надежности, вероятности опоздания для маршрутов от числа клиентов, анализ которых показал высокое качество маршрутов, рекомендуемых ГиИМАС.

Одна из графических зависимостей, отображающая причинно-следственную связь числа ситуаций (в процентах), в которых возникает синергетический эффект и, как следствие, коллективное решение (решение ЛПР) оказывается лучшим, чем любое индивидуальное решении эксперта, от количества клиентов, дана на рис. 3. Видно, что качество принимаемых решений ГиИМАС с нейтральными агентами выше, чем ГиИМАС других архитектур. Это прямое следствие того, что в ГиИМАС с нейтральными агентами вероятность возникновения синергетического эффекта выше. Однако чем меньше размерность задачи, тем менее влияет этот эффект на качество решения.

Следует отметить, что эффективность ГиИМАС при любом взаимодействии агентов выше, чем без такового.

Анализ технико-эксплуатационных параметров показывает, что в условиях эксперимента время решения сложной транспортно-логистической задачи составило 2–30 мин, а качество решений подтверждено экспериментально и экспертами. По итогам практического использования программного продукта, реализующего ГиИМАС на двух объектах, средняя суммарная себестоимость доставки грузов в день сократилась на 7,2%, средняя суммарная длительность доставки в день — на 12,13%, среднее время построения маршрутов в день уменьшилось на 23,14%.

# 6 Заключение

В рамках парадигмы синергетического искусственного интеллекта рассмотрен подход к со-

зданию КСППР, моделирующей СППР и коллективные эффекты в них, приведена ее модель и функциональная структура. Система динамически перестраивает алгоритм своего функционирования, каждый раз при работе над сложной задачей вырабатывая релевантный ей метод решения. Метод моделирования самоорганизации СППР на основе анализа целей агентов, по сути, позволяет разнообразию условий и ситуаций решения сложных задач коллективом сопоставить не один-единственный инструмент, а множество динамично синтезируемых и изменяемых интегрированных моделей. Это открывает компьютерным системам управления перспективу для решения сложных задач, с которыми сейчас могут справиться только коллективы экспертов.

Вычислительные эксперименты на ГиИМАС с самоорганизацией сделали контрастными новые знания о релевантности архитектур ГиИМАС различным ситуациям решения задачи в СППР.

Сравнение ГиИМАС и аналогичной гибридной интеллектуальной системы [5] для решения сложной транспортно-логистической задачи показало, что моделирование эффекта самоорганизации позволило динамически синтезировать метод решения сложных задач на основе базовых методов — знаний агентов:

(1) при решении сложной транспортно-логистической задачи вырабатывается релевантный ей метод, что повышает качество решения;

(2) стало возможным решать задачи с более высокими оценками сложности моделирования [5];

(3) сокращается трудоемкость проекта за счет отказа от настройки межагентных связей;

(4) хотя, по оценке, сложность рассмотренной тестовой задачи выше сложности задачи, решаемой ГиИС, скорость получения результата ГиИМАС выше примерно в 2–4 раза.

# Литература


1. *Трахтенгерц Э. А.* Компьютерная поддержка принятия решений. — М.: Синтег, 1998.
2. *Колесников А. В.* Гибридные интеллектуальные системы. Теория и технология разработки / Под ред. А. М. Яшина. — СПб.: СПбГТУ, 2001.
3. *Freud S.* Group psychology and the analysis of the Ego. — New York: Liveright Publishing, 1922.
4. *Lewin K.* Resolving social conflicts: Selected papers on group dynamics. — New York: Harper & Row, 1948.
5. *Колесников А. В. Кириков И. А., Листопад С. В., Румовская С. Б., Доманицкий А. А.* Решение сложных задач коммивояжера методами функциональных гибридных интеллектуальных систем / Под ред. А. В. Колесникова. — М.: ИПИ РАН, 2011.
6. *Поспелов Д. А.* Десять «горячих точек» в исследованиях по искусственному интеллекту // Интеллектуальные системы, 1996. Т. 1. Вып. 1-4. С. 47–56.
7. *Почебут Л. Г., Чикер В. А.* Организационная социальная психология: Уч. пособие. — СПб.: Речь, 2002.
8. *Майерс Д.* Социальная психология. — СПб.: Питер, 1997.
9. *Кричевский Р. Л., Дубовская Е. М.* Социальная психология малой группы. — М.: Аспект Пресс, 2001.
10. Социальная психология / Под ред. А. Л. Журавлева. — М.: ПЕР СЭ, 2002.
11. *Тарасов В. Б.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. — М.: Эдиториал УРСС, 2002.
12. *Листопад С. В.* Интеллектуальная система моделирования коллективного принятия решений для сложной транспортно-логистической задачи: Дисс. . . . канд. техн. наук. — М.: ИПИ РАН, 2012. 151 с.
13. *Brockett R. W.* Hybrid models for motion control systems. Technical Report CICS-P-364. — Massachusetts Institute of Technology, Center for Intelligent Control Systems, 1993.
14. *Рыбина Г. В.* Проектирование систем, основанных на знаниях. — М.: МИФИ, 1997.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 70–80

# СЕМАНТИКА АСПЕКТНО-ОРИЕНТИРОВАННОГО МОДЕЛИРОВАНИЯ ДАННЫХ И ПРОЦЕССОВ

С. П. Ковалёв[1]

**Аннотация:** Предложен подход к унификации технологий аспектно-ориентированного программирования (АОП) на семантическом уровне путем формализации основных понятий АОП средствами теории категорий. Технология АОП описывается категорией аспектно-ориентированных моделей программ (АО-моделей) и их системных взаимосвязей, снабженной функтором выделения аспектной структуры (разметки моделей классами задач). Связывание аспектно-ориентированных программ формализуется универсальной конструкцией в этой категории. Построены и проанализированы формальные технологии АОП, применение которых позволяет снижать затраты на моделирование данных и сценариев исполнения процессов. Строго сформулировано и обосновано условие существования связывания для сценарных моделей программных систем.

**Ключевые слова:** аспектно-ориентированное программирование; теория категорий; аспектное связывание

## 1 Введение

Традиционные технологии программирования, в том числе структурные и объектно-ориентированные, предписывают собирать сложные системы из модулей, предназначенных для выполнения различных функциональных задач и взаимодействующих между собой через фиксированные интерфейсы. Однако в практике создания систем постоянно возникают классы задач, не поддающиеся локализации в рамках модулей. Они рассеиваются (scatter) по разным модулям, пересекают (crosscut) их границы, перемешиваются (tangle) с реализацией других задач. Поэтому их автоматизация средствами «модульных» технологий сопряжена со значительными затратами труда: качество результата зависит не только от качества средства программной реализации задачи, но и от полноты его проникновения во все части системы. Примеры можно найти как среди функциональных задач (ведение паспорта объекта автоматизации, верификация данных), так и среди программно-технических (ведение журналов функционирования системы, защита информации и т. д.).

Эффективное создание систем, содержащих такие рассеянные задачи, является целью АОП — новой парадигмы, предложенной Г. Кишалесом и его коллегами в конце 1990-х гг. [1]. Реализация рассеянной задачи в АОП оформляется как аспект — особая программная единица, код которой автоматически вставляется в код других единиц в точках, явно задаваемых внешним образом. Однако на практике АОП применяется значительно реже, чем модульные подходы, поскольку отсутствует единое непротиворечивое понимание его методологической основы [2]. На семантическом уровне неясно, как рационально выделять и комплексировать аспекты в программных системах и их моделях. Существующие технологии АОП предлагают лишь частные решения, специфичные для частных парадигм программирования.

Сходные проблемы препятствовали развитию объектно-ориентированного подхода, пока не был создан универсальный язык моделирования UML (Unified Modeling Language), позволивший освободить процесс объектной декомпозиции от ограничений частных языков программирования [3].

В связи с этим целью настоящей работы стало построение универсальной семантики АОП, не зависящей от выбора парадигмы программирования. Для этого был привлечен аппарат теории категорий, поскольку он позволяет единообразно и компактно описывать разнородные технологии инженерии программного обеспечения с позиций системного анализа [4]. С его помощью удалось компактно выразить двухуровневый характер технологий АОП — составление программы из модульной основы и аспектной структуры. Сборка программ из аспектов формализована универсальной категорной конструкцией. В качестве приложения этого подхода предложена единая концепция технологии АОП, позволяющей снижать затраты на моделирование данных и сценариев исполнения процессов — ключевых составляющих широкого класса программных систем.

[1]Институт проблем управления им. В. А. Трапезникова Российской академии наук, kovalyov@nm.ru

## 2 Аспектно-ориентированное программирование

Классы задач, пересекающие границы единиц модульной архитектуры, хорошо известны программистам, использующим широко распространенные алгоритмические языки. Практически в любой программе исполнение основных функций перемешивается с обращениями к программно-техническим задачам, таким как журналирование, кэширование, защита информации и т. п. Они глубоко погружены в контекст своего исполнения, поэтому их реализацию трудно оформлять единицами модульной архитектуры, контекст которых передается через фиксированный интерфейс («обобщенными процедурами», в терминологии Г. Кишалеса). Приходится дублировать программный код, реализующий рассеянные задачи, в различных местах обращения к ним, чрезвычайно повышая трудоемкость модификации программы. Даже если искусственно оформить рассеянные задачи процедурами, то перемежающиеся обращения к ним, нагруженные передачей контекста, делают текст программы трудным для понимания и сопровождения.

В рамках АОП предлагается оформлять реализации рассеянных задач в виде аспектов — особых программных единиц, места обращения к которым задаются в отдельной спецификации, внешней по отношению к вызывающей их («базовой») программе. Сборка системы из аспектов заключается во вставке их программного кода в базовый код в этих местах, в результате чего они получают полный доступ к контексту, обходя ограничения модульного интерфейса доступа. Эта процедура выходит за рамки традиционной компоновки модулей (linking), не позволяющей модифицировать поток исполнения программы, поэтому она называется связыванием (weaving). Связывание может выполняться как на этапе компиляции, путем генерации программного кода со вставками согласно спецификации мест обращения к аспектам, так и в процессе исполнения, путем вызова предварительно скомпилированных аспектов при достижении соответствующих мест. Поэтому связывание может способствовать снижению затрат не только в условиях рассеяния задач, но и в других случаях, когда требуется изменить поведение программного модуля внешним «нештатным» образом. Например, так можно реализовать модули, предназначенные для многократного использования в конфигурациях, различающихся наличием/отсутствием нескольких изменчивых задач, оформляемых как аспекты [5].

Еще одной областью такого применения связывания является интеграция унаследованных модулей (legacy), не доступных в исходных кодах и потому не допускающих модификацию путем редактирования [6].

Приведем наглядный пример с использованием языка AspectJ [7] — аспектно-ориентированного расширения языка Java. Предположим, что при выполнении некоторой программы требуется распечатывать все вызовы методов, наименования которых начинаются со слова init (инициализировать). Ручная реализация такой задачи включает поиск этого слова в тексте программы, умозрительное отделение вызовов методов (от наименований переменных, комментариев и др.) и вписывание обращения к функции печати после них. Компилятор AspectJ выполняет всю эту работу автоматически, получив на вход аспект следующего вида:

```
public aspect methodCallLogging {
    // Регулярное выражение,
    // задающее точки вставки
    // кода аспекта в исходную программу
    pointcut methodCalled():
      execution(public * init*(..));
    // Действие, вставляемое после
    // каждой точки
    after(): methodCalled() {
      System.out.println("Method call: " +
           thisJoinPoint.getSignature());
    }
}
```

Этот пример позволяет увидеть и основную концептуальную проблему классического АОП — зависимость результата связывания от синтаксической структуры программы, а не от семантической структуры ее предметной области. Так, пример не обеспечит журналирование вызова метода инициализации в случае, если он назван begin() вместо init(). В литературе эта проблема называется «композиционной хрупкостью» (composition fragility) [8]: многоаспектная система может радикально измениться и даже развалиться на части при незначительном изменении синтаксиса базы, хотя семантика остается неизменной. Этим вызвано удручающее однообразие типичных областей применения АОП: они не выходят за рамки программно-технических задач [2]. Не хватает эффективных типовых решений по реализации семантически богатых функциональных рассеянных задач, таких как ведение паспорта объекта автоматизации, оповещение участников процессов о ходе их выполнения, оперативная оценка эффективности процессов, проверка правильности действий пользователей и компонентов системы, перевод информации на различные языки и в различные форматы.

Полноценная поддержка функциональных аспектов требует распространения аспектно-ориентированного подхода на весь жизненный цикл программного обеспечения — от формирования требований до сопровождения готового изделия [9]. В начале жизненного цикла аспекты естественным образом появляются в виде классов задач (concerns), присутствующих одновременно во многих требованиях. Такие аспекты называются ранними (early aspects) [8]. По своей природе они вполне соответствуют значению слова «аспект» в обыденном языке: «(*лат*. aspectus — вид, взгляд) точка зрения, взгляд на что-нибудь» [10]. В литературе описан ряд частных подходов к анализу и моделированию ранних аспектов, однако они не имеют единой семантической базы, которая позволила бы обеспечить их совместимость друг с другом, прямой переход к реализации средствами АОП, формальную проверку корректности [8]. Ни один из подходов не способен претендовать на роль канона аспектно-ориентированного моделирования.

Как свидетельствует история инженерии программного обеспечения, в этих условиях необходима абстрактная семантическая модель, не зависящая от выбора языков и технологий программирования. Наиболее прямым операциональным выражением целевой установки АОП является пометка фрагментов программ классами задач, на решение которых они направлены. Действительно, мотивация создателей АОП состояла в нехватке программных конструкций для явного разделения ответственности (separation of concerns) в исходном коде [1]. Разметка классами задач образует аспектную структуру программы, сохранение которой по ходу процесса разработки позволяет избежать главной проблемы, вызванной рассеянием задач, — «растворения» аспектов в контексте и утраты их идентичности. Технологии АОП отличаются способами разметки, но сходятся в стремлении обеспечить сквозную трассируемость, т. е. возможность однозначно установить, для чего в программу включен тот или иной фрагмент. Недостаточная поддержка трассирования, предоставляемая большинством распространенных инструментов моделирования и составления программ, является главным мотивом привлечения технологий типа АОП [8]. Поэтому предлагаемая семантика АОП опирается на концепцию трассирования, в ходе которого процедуры сборки программных систем отражаются на уровне классов решаемых ими задач.

В заключение раздела отметим, что в традиционной инженерии материальных систем раздельное решение рассеянных задач является давно устоявшейся практикой. Рассмотрим в качестве примера проектирование зданий — область, из которой инженеры по программному обеспечению заимствовали много идей. Модульную архитектуру здания составляют подъезды, этажи, комнаты, а аспектную — интенсивно пересекающие их системы освещения, водоснабжения, отопления и др. [9]. Проект каждого из таких аспектов документируется в форме отдельного плана и отдается на реализацию отдельной бригаде специалистов. Связывание аспектов по ходу строительства, включая разрешение всевозможных технологических и процедурных конфликтов, входит в число рутинных задач прораба.

## 3 Формальные технологии аспектно-ориентированного программирования

В качестве математического аппарата для строгой записи семантики АОП была привлечена теория категорий. Предполагается, что читатель знаком с ее базовыми понятиями; их определения можно найти, например, в книгах [11, 12] (в [12] объекты и морфизмы категории $C$ кратко называются $C$-объектами и $C$-морфизмами соответственно). Теоретико-категорный подход к формализации программирования разрабатывается начиная с 1970-х гг. [4]. Каждому компоненту или системе сопоставляется абстрактный объект, а каждому действию по интеграции компонента в систему — морфизм, т. е. абстрактный аналог функции, преобразующей объект-область (компонент) в объект-кообласть (систему). Подчеркнем, что для любого компонента $C$ и любой системы $S$, как правило, указывается не только возможность (или невозможность) интеграции $C$ в $S$, но и совокупность всех различных способов интеграции (вставка, разрешение ссылок, перекомпоновка и т. д.), образующая множество морфизмов $\mathrm{Mor}(C, S)$. Композиция морфизмов отвечает конструированию многошаговых действий (процессов), причем их результат не зависит от порядка прослеживания шагов (свойство ассоциативности). Имеются тождественные морфизмы, означающие «ничегонеделание». Процедуры сборки систем из компонентов описываются диаграммами — ориентированными графами, вершины которых помечены объектами, а ребра — морфизмами. Таким образом, получается *формальная технология программирования* — категория, конструкции в которой описывают приемы комплексирования систем.

Среди конструкций, важных для дальнейшего изложения, отметим терминальный объект — эле-

ментарную модель, лишенную какой-либо внутренней структуры. Любая модель может быть формально интегрирована в нее, причем единственным способом — путем полного «сокрытия» своей структуры. В свою очередь, морфизм вида $e : \mathbf{1} \to S$, где $\mathbf{1}$ — терминальный объект, задает элемент модели $S$ [11]. Например, в категории **Set**, состоящей из всех множеств и всех отображений, терминальным объектом служит одноэлементное множество. Любое множество отображается в него единственным образом, а любое отображение его в некоторое множество выделяет один из элементов последнего.

В условиях применения АОП комплексирование систем происходит согласованно на двух уровнях: моделей программ и их аспектных структур. Отражение действий по интеграции на уровне классов задач, образующих аспектную структуру, естественным образом формализуется функтором, извлекающим аспектную структуру из программы [13].

**Определение 1.** Пусть заданы:

- категория AO, объектами которой служат формальные AO-модели, а морфизмами — действия по интеграции (программных систем);
- функтор str из AO в категорию, обозначаемую STR, объектами которой служат аспектные структуры AO-моделей, а морфизмами — действия по интеграции аспектов.

*Формальной технологией АОП* называется пара $\langle \mathrm{AO}, \mathrm{str} \rangle$. Формальная технология АОП называется *элементарной*, если категория AO имеет терминальный объект и функтор str сохраняет его (т. е. переводит в терминальный STR-объект). □

Конечно, не всякая пара, состоящая из категории и действующего из нее функтора, пригодна в качестве формализации какой-либо существующей (или возможной в будущем) технологии АОП — для этого необходимо наложить ряд громоздких дополнительных условий. Наиболее простое из них состоит в том, что любая аспектная структура $Q$ должна быть реализуема — должна существовать AO-модель $A$ такая, что $\mathrm{str}(A) = Q$. Однако результаты настоящей работы не зависят от таких условий, поэтому, чтобы облегчить чтение, они здесь не приводятся. Интересующихся читателей отсылаем к работам [14, 15].

При описании частных методов программирования объектами формальных технологий служат модели программ того или иного вида: алгебраические спецификации, графы, термы лямбда-исчисления и т. д. Простой пример появляется в формальном подходе к инженерии данных. Наиболее общая модель массива данных представляет собой множество, состоящее из хранящихся в нем информационных элементов. Действие по интеграции массивов — это в точности любое отображение множеств, поскольку каждому элементу массива-компонента сопоставляется единственный элемент массива-системы. Конкретные информационные технологии оперируют со специальными видами множеств; например, таблица в реляционной базе данных представляет собой подмножество прямого произведения основных множеств типов атрибутов (domains, см., например, [16]).

Интеграция двух таблиц выполняется путем формирования внешнего ключа — отображения множества всех записей одной таблицы во множество записей второй. Таким образом, морфизмы таких множеств и составленные из них диаграммы описывают конструкции реляционной алгебры. При моделировании объектно-ориентированных баз данных множества сопоставляются классам, а отображения — декларации атрибутов и наследованию [17]. Применение АОП в инженерии данных можно смоделировать без ограничения общности: в соответствии с соображениями, изложенными во введении, элементы массивов помечаются именами классов задач, оперирующих с ними [18]. Например, в системах технологического управления ведется общий реестр оборудования, включающий технологические узлы, измерительные приборы и исполнительные механизмы, вычислительные устройства и др. В целях разделения единиц оборудования по их назначению в реестр добавляется специальный атрибут-метка, имеющий перечислимый тип (классификатор оборудования).

По существу (с точностью до изоморфизма), разметка является отношением эквивалентности, классы которого отвечают отдельным аспектам, так что аспектная структура представляет собой фактор-множество. Действиям по интеграции отвечают в точности все отображения, совместимые с аспектной структурой, т. е. сохраняющие разметку. Получается формальная технология АОП $\mathrm{ADM} = \langle \mathbf{Equ}, \mathrm{fs} \rangle$, где **Equ** — категория всех множеств с отношением эквивалентности и всех их гомоморфизмов, fs: $\mathbf{Equ} \to \mathbf{Set} : \langle S, \sim \rangle \mapsto S/\sim$ — функтор вычисления фактор-множества [13]. Эта формальная технология элементарна, поскольку одноэлементное множество с тривиальным отношением эквивалентности является терминальным **Equ**-объектом.

Немного сложнее технологии моделирования данных выглядит формальная технология АОП, отвечающая моделированию процессов функционирования систем. Строительным материалом для

моделей процессов служат сценарии — последовательности действий и взаимодействий, происходящих при определенных условиях, изложенные без предложений с «если» и ветвления [19]. (Здесь слово «последовательность» следует понимать в широком смысле, так как сценарий может содержать взаимно независимые параллельные события, ни одно из которых не следует за другим.) Поэтому АО-моделью здесь служит помеченный сценарий — множество событий, частично упорядоченное причинно-следственной связью и размеченное классами порождающих их задач [14, 20]. При интеграции сценариев сохраняется и порядок, и разметка. Примером служит подключение единого журнала событий: журнал формально задается множеством вещественных чисел $\mathbb{R}$ с естественным линейным порядком, описывающим стрелу физического времени, и одноэлементной разметкой (задача «журналирование»), а регистрация в нем событий, образующих сценарий $X$, — гомоморфизмом вида $t: X \to \mathbb{R}$. Таким образом получается элементарная формальная технология АОП $\mathrm{ASM} = \langle \mathbf{PosEqu}, \mathrm{fs} \circ \mathrm{equ} \rangle$, где **PosEqu** — категория всех частично упорядоченных множеств с отношением эквивалентности и всех их гомоморфизмов, $\mathrm{equ}: \mathbf{PosEqu} \to \mathbf{Equ}: \langle S, \leq, \sim \rangle \mapsto \langle S, \sim \rangle$ — функтор, «забывающий» частичный порядок [13]. Подобный подход к моделированию сценариев был предложен еще в 1980-х гг. [21], однако природа меток и способы их синтеза оставались неясными, поскольку они не рассматривались в контексте АОП. Фактически моделирование поведения систем помеченными сценариями выступает в роли операционной семантики АОП, поскольку, как будет показано ниже, оно непосредственно отражает концепцию связывания аспектов (weaving). Его частный случай, охватывающий упрощенный вариант АОП, известен как трассовая семантика аспектов [22]. Частные технологии инженерии процессов оперируют с разнообразными классами сценариев (подклассами в Ob **PosEqu**), позволяя записывать их в различных специализированных нотациях: графических, алгебраических, гипертекстовых, сетей Петри и др.

Заметим, что формальная технология моделирования данных ADM является «корефлексивной подтехнологией» в ASM: функтор полного вложения $\mathrm{equ}^*: \mathbf{Equ} \hookrightarrow \mathbf{PosEqu}: \langle S, \sim \rangle \mapsto \langle S, =, \sim \rangle$ является левым сопряженным и одновременно правым обратным к equ, так что $(\mathrm{fs} \circ \mathrm{equ}) \circ \mathrm{equ}^* = \mathrm{fs}$. Благодаря этому можно обеспечить полную трассируемость данных к процессам, снабжая элементы данных метками классов задач, в процессе выполнения которых они порождаются или модифицируются. Пример такого совместного моделирования данных и процессов будет приведен в конце разд. 4.

Основываясь на подходах, описанных в литературе по теории АОП, можно построить ряд более частных и более сложных формальных технологий. Однако в настоящей работе ограничимся двумя вышеописанными, поскольку основной объем работ по созданию широкого класса программных систем относится к моделированию данных и процессов.

# 4 Аспекты и связывание

Рассеяние задач приводит к повышению затрат на проектирование и эксплуатацию программных систем во многом из-за того, что сборка системы может разрушить аспектную структуру ее компонентов. Избежать этого можно путем применения действий по интеграции, позволяющих идентифицировать аспектную структуру компонента в структуре системы путем трассирования. Наилучшим же с точки зрения АОП является действие, не вызывающее существенных изменений в аспектной структуре. Зафиксируем произвольную формальную технологию АОП $\mathrm{AS} = \langle \mathrm{AO}, \mathrm{str} \rangle$. На языке теории категорий трассирование действия (АО-морфизма) $f: X \to S$ в направлении от результата к источнику естественно описать АО-морфизмом $g: S \to X$, левым обратным к $f$, т. е. удовлетворяющим условию $g \circ f = 1_X$. Отметим, что любое трассируемое действие является регулярным мономорфизмом — категорным аналогом вложения. Если, в свою очередь, морфизм $f$ пригоден для трассирования морфизма $g$, т. е. если $f \circ g = 1_S$, то изменение, вызываемое действием, можно считать несущественным: как обычно в теории категорий, несущественному изменению отвечает изоморфизм — морфизм, обратимый как слева, так и справа [12, разд. 3.4]. Эти соображения служат мотивировкой для нижеследующего определения.

**Определение 2.** Аспектно-ориентированный морфизм $f$ называется *аспектным*, если $\mathrm{str}(f)$ является сечением (т. е. имеет левый обратный), и *изоаспектным*, если $\mathrm{str}(f)$ является изоморфизмом. □

Например, в технологиях АОП, построенных в разд. 3, аспектными являются отображения с непустой областью, факторизующиеся в инъекции. Они не «склеивают» метки и потому допускают однозначное трассирование на уровне аспектной структуры.

Аспектом (aspect) называется элементарный строительный блок аспектно-ориентированной программы, реализующий отдельный класс задач.

Как указывалось в разд. 2, технология АОП нацелена на сохранение идентичности аспектов в составе программы, поэтому их аспектная структура не может быть разрушена при интеграции в любую систему. Это свойство и составляет формальное определение аспекта. Подчеркнем, что оно выходит за рамки классического АОП, где аспект обязательно должен вставляться в некоторую базовую программу посредством связывания: предлагаемая семантика позволяет избавиться от заложенного классиками неявного порочного круга. Например, будет доказано, что в элементарной формальной технологии АОП аспекты — это в точности все АО-модели, аспектная структура которых элементарна.

**Определение 3.** Аспектно-ориентированная модель $A$ называется *аспектом*, если любой АО-морфизм с областью $A$ является аспектным. □

**Предложение 1.** Следующие утверждения эквивалентны для любой АО-модели $A$:

(i) $A$ является аспектом;

(ii) $A$ изоморфно аспекту;

(iii) существует изоаспектный морфизм, направленный из некоторого аспекта в $A$;

(iv) существует сечение, направленное из $A$ в некоторый аспект.

Доказательство.

(i) $\Rightarrow$ (ii). Тождественный морфизм $1_A : A \to A$ является изоморфизмом.

(ii) $\Rightarrow$ (iii). Любой АО-изоморфизм изоаспектен.

(iii) $\Rightarrow$ (i). Если $B$ — аспект и $u : B \to A$ — АО-морфизм, то согласно определению 3 $\mathrm{str}(f \circ u)$ является сечением для любого АО-морфизма $f$ с областью $A$. В частности, если $u$ изоаспектен, то STR-морфизм $\mathrm{str}(f) = \mathrm{str}(f \circ u) \circ \mathrm{str}(u)^{-1}$ является сечением.

(i) $\Rightarrow$ (iv). Тождественный морфизм $1_A$ является сечением.

(iv) $\Rightarrow$ (iii). Рассмотрим сечение $s : A \to B$, где $B$ — аспект. Пусть $s' : B \to A$ — АО-морфизм такой, что $s' \circ s = 1_A$. STR-морфизм $\mathrm{str}(s')$ имеет как правый обратный (это $\mathrm{str}(s)$), так и левый обратный (определение 3), следовательно, он является изоморфизмом, т. е. $s'$ изоаспектен. □

**Предложение 2.** Если формальная технология AS элементарна, то следующие утверждения эквивалентны для любой АО-модели $A$:

(i) $A$ является аспектом;

(ii) $\mathrm{str}(A)$ является терминальным STR-объектом;

(iii) существует аспектный морфизм, направленный из $A$ в некоторый аспект.

Доказательство. Напомним, что терминальный объект традиционно обозначается через $\mathbf{1}$. Заметим, что в любой категории: (а) любой морфизм вида $i : \mathbf{1} \to X$ является сечением (морфизм $!_X : X \to \mathbf{1}$ является левым обратным к нему), поэтому (б) любое сечение вида $!_X : X \to \mathbf{1}$ является изоморфизмом.

(i) $\Rightarrow$ (ii). Если $A$ — аспект, то ввиду (б) морфизм $\mathrm{str}(!_A)$ является изоморфизмом.

(ii) $\Rightarrow$ (i). Если $\mathrm{str}(A)$ — терминальный объект, то в силу (а) любой морфизм с областью $A$ является аспектным.

(i) $\Rightarrow$ (iii). Тождественный морфизм $1_A$ является аспектным.

(iii) $\Rightarrow$ (ii). Если $B$ — аспект (т. е., как уже доказано, $\mathrm{str}(B)$ — терминальный объект) и $u : A \to B$ — аспектный АО-морфизм, то $\mathrm{str}(u)$ представляет собой сечение, направленное в терминальный объект. Поэтому ввиду (б) $\mathrm{str}(A)$ также является терминальным объектом. □

Рассмотрим процедуру связывания системы из аспектов. Напомним, что в классическом АОП оно состоит в подключении программы $W$, называемой советом (advice), к базовой программе (base) $B$ в заданных местах, называемых точками соединения (join points). Каждый раз, когда при исполнении базовой программы встречается точка соединения, вызывается совет. Поэтому последний обычно выглядит как блок программного кода, охраняемый (guarded) условием, идентифицирующим точку соединения; начало блока служит точкой вызова совета (entry point). Пример такого блока на языке AspectJ был приведен в разд. 2. Таким образом, инструмент связывания (weaver) принимает на вход две спецификации:

(1) описание точек соединения в базовой программе, или срез (pointcut);

(2) описание точек вызова совета в точках соединения.

При связывании сначала (виртуально) создается достаточное количество копий совета, по одной на каждую точку соединения, с маркировкой соответствующих им точек вызова. Далее эти точки «склеиваются» друг с другом так, чтобы не разрушить аспектную структуру базы и совета. Для формальной записи правил связывания привлекается дополнительная АО-модель $C$, называемая связкой (connector, см. [23]), которая интегрируется с базой в точках соединения, а с советом — в точках вызова. В технологиях типа AspectJ в роли

связки выступает регулярное выражение, выделяющее в тексте базовой программы синтаксические единицы, образующие срез (конструкция pointcut). Соответствие точек соединения точкам вызова задается парой АО-морфизмов $j : B \leftarrow C \rightarrow W : e$ (здесь наглядно проявляется отличие связывания от модульной компоновки, формализуемой одношаговым действием вида $l : M \rightarrow S$, где $M$ — модуль, $S$ — система).

Как легко видеть на примере формальной технологии АОП для моделирования сценариев, первый шаг связывания может быть формализован как построение произведения $C \times W$, а второй — кодекартова квадрата (стандартная конструкция склеивания элементов множества). Эти операции должны быть естественными относительно вычисления аспектной структуры, чтобы получилось связывание разметок: указанные универсальные конструкции должны сохраняться функтором str.

В литературе рассмотрены два частных случая вычисления связывания как кодекартова квадрата того или иного вида: когда метки трактуются как роли [24] и когда аспекты задаются как инварианты поведения программ, описанных алгебраическими спецификациями [25]. Данное выше определение, напротив, имеет общий характер, и оно отражает ряд интуитивно ожидаемых свойств связывания (например, возможность трассируемого включения базы в результат).

**Определение 4.** *Аспектным связыванием* пары АО-морфизмов $j : B \leftarrow C \rightarrow W : e$, где $B$ называется базой, $W$ — советом, $C$ — связкой, называется кодекартов квадрат пары $j : B \leftarrow C \rightarrow C \times W : \langle 1_C, e\rangle$ (схемы связывания), если он существует (в частности, существует произведение $C \times W$) и функтор str сохраняет как произведение $C \times W$ (переводит его в произведение), так и этот кодекартов квадрат (переводит его в кодекартов квадрат). *Результатом связывания* называется вершина кодекартова квадрата, обозначается через $j \bowtie e$ (рис. 1). □

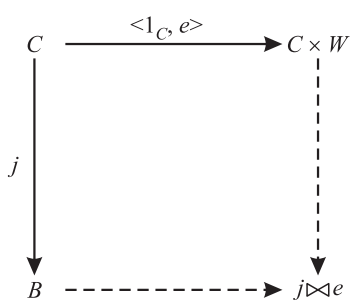



**Рис. 1** Кодекартов квадрат связывания

**Предложение 3.** Для любой пары

$$j : B \leftarrow C \rightarrow W : e\,,$$

обладающей связыванием, справедливы следующие утверждения:

(i) результат связывания определяется единственным образом (с точностью до изоморфизма);

(ii) существует обратимое слева (в частности, аспектное) вложение $b : B \hookrightarrow (j \bowtie e)$;

(iii) если результат связывания является аспектом, то и база является аспектом;

(iv) если $j$ — изоморфизм, то $j \bowtie e \cong B \times W$.

Доказательство.

(i). Вытекает из универсальности (ко)пределов.

(ii). Имеем $\pi_C \circ \langle \mathbf{1}_C, e\rangle = \mathbf{1}_C$, где $\pi_C : C \times W \rightarrow C$ — проекция, так что $\langle \mathbf{1}_C, e\rangle$ — сечение. В свою очередь, как легко проверить, ребро кодекартова квадрата, параллельное сечению, само является сечением (см. двойственное утверждение в [11, предложение 11.18]).

(iii). Вытекает из утверждения (ii) и предложения 1 (эквивалентность (i) ⇔ (iv)).

(iv). Ребро кодекартова квадрата, параллельное изоморфизму, само является изоморфизмом. □

Утверждение (ii) предложения 3 позволяет описать многошаговое связывание взаимно независимых советов с общей базой: если имеются связываемые пары

$$j : B \leftarrow C \rightarrow W : e\,;$$
$$j' : B \leftarrow C' \rightarrow W' : e'\,,$$

то их можно естественным образом собрать в одно целое путем связывания пары $b \circ j' : (j \bowtie e) \leftarrow C' \rightarrow W' : e'$. Покажем, что результат здесь не зависит от порядка привязки советов.

**Предложение 4.** Если пары $j : B \leftarrow C \rightarrow W : e$, $j' : B \leftarrow C' \rightarrow W' : e'$, $b \circ j' : (j \bowtie e) \leftarrow C' \rightarrow W' : e'$, $b' \circ j : (j' \bowtie e') \leftarrow C \rightarrow W : e$ обладают связыванием, то $(b \circ j') \bowtie e' \cong (b' \circ j) \bowtie e$.

Доказательство. Пусть *AO*-морфизмы $u$, $v$, $w$ таковы, что соотношения $u \circ \langle \mathbf{1}_C, e\rangle = b \circ j$ и $v \circ \langle \mathbf{1}_C, e\rangle = w \circ (b' \circ j)$ задают кодекартовы квадраты. Тогда существует морфизм $q : (j \bowtie e) \rightarrow (b' \circ j) \bowtie e$ такой, что $q \circ u = v$ и $q \circ b = w \circ b'$. В силу общего утверждения, двойственного к известной лемме о декартовых квадратах [12, предложение 11.10 (2)], второе из этих равенств определяет кодекартов квадрат пары $b : (j \bowtie e) \hookleftarrow B \hookrightarrow (j' \bowtie e') : b'$ с вершиной $(b' \circ j) \bowtie e$. Рассуждая аналогично, по-

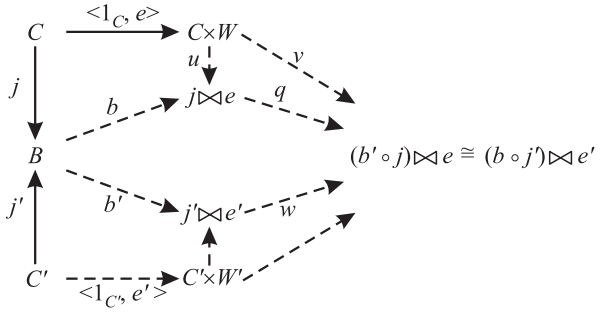



**Рис. 2** К доказательству предложения 4

лучаем, что объект $(b \circ j') \bowtie e'$ является вершиной кодекартова квадрата той же пары (рис. 2). □

Эффект перемешивания классов задач при связывании ярко проявляется, если совет является достаточно мелкой единицей аспектной архитектуры, например аспектом. База связывания в некоторой степени поглощает такую единицу, затрудняя ее идентификацию в составе системы. Для формальной характеризации таких единиц введем следующее понятие. Будем называть объект некоторой категории *частично терминальным*, если из любого объекта в него имеется не более одного морфизма. Объект, изоморфный частично терминальному, сам является частично терминальным. Ясно, что любой терминальный объект является частично терминальным. В любой категории предпорядка все объекты частично терминальны.

**Предложение 5.** Для любой пары

$$j : B \leftarrow C \rightarrow W : e\,,$$

обладающей связыванием, справедливы следующие утверждения:

(i) если $W$ — частично терминальный АО-объект, то $j \bowtie e \cong B$;

(ii) если $str(W)$ — частично терминальный STR-объект, то $\mathrm{str}(j \bowtie e) \cong \mathrm{str}(B)$.

Доказательство. Сначала заметим, что если $X$ — частично терминальный объект в произвольной категории и в ней существует морфизм $f : Y \rightarrow X$, то объект $Y$ представляет собой произведение $Y \times X$ с проекциями $1_Y$ и $f$ (с точностью до изоморфизма).

(i). Если $W$ — частично терминальный АО-объект, то, как было только что установлено, $\pi_C$ — изоморфизм, поэтому $\langle 1_C, e\rangle$ — изоморфизм.

(ii). Поскольку $\mathrm{str}(C \times W) = \mathrm{str}(C) \times \mathrm{str}(W)$, аналогично предыдущему получаем, что $\langle 1_C, e\rangle$ — изоаспектный АО-морфизм. □

Рассмотрим аспекты и связывание в формальных технологиях АОП, описанных в разд. 2. Согласно предложению 2, здесь аспект — это в точности любое непустое множество, все элементы которого помечены одной и той же меткой. Отношение эквивалентности, задающее разметку, разбивает АО-модель на совокупность непересекающихся аспектов. Частично терминальными объектами являются только пустое и одноэлементное множество.

В технологии моделирования данных ADM связывание существует всегда, поскольку категория **Equ** (ко)полна, а функтор факторизации fs сохраняет произведения (проверяется непосредственно) и копределы (поскольку он имеет правый сопряженный $\mathrm{fs}_* : \mathbf{Set} \hookrightarrow \mathbf{Equ} : S \mapsto \langle S, =\rangle$).

Результат связывания получается из раздельного объединения $B \amalg (C \times W)$ путем амальгамирования — факторизации по отношению эквивалентности, порожденному множеством пар $\{(j(x), (x, e(x)))|x \in C\}$ [11, разд. 3.14]. Связывание имеет прозрачный смысл для реляционных таблиц: здесь спецификация связывания описывает добавление к базе атрибута типа «список ссылок на совет» (отношение вида «многие ко многим»), с использованием связки в качестве источника ссылочных ключей. Добавление атрибута является неразрушающим в том смысле, что структура базовой таблицы не меняется, поэтому оно не требует значительных затрат. Такой прием характерен для аспектно-ориентированной разработки информационных моделей сложных объектов управления, элементам которых свойственно вступать в многообразные и изменчивые отношения [18]. Здесь метки связки обозначают функции (роли) отношений между сущностями, описываемыми в таблицах базы и совета. Отношения, помеченные ролями, возникают между видами объектов и средств управления (онтологические отношения), их типами и марками (нормативные) и экземплярами (фактические). Например, на онтологическом уровне описываются возможности установки различных видов приборов, реализующих те или иные функции управления, на различных видах зданий и сооружений, на нормативном — правила выбора марок приборов для оснащения зданий различных серий, на фактическом — результаты работ по монтажу конкретных приборов в конкретных зданиях.

В формальной технологии моделирования сценариев ASM связывание существует не всегда, хотя **PosEqu** и (ко)полна: функтор equ имеет левый сопряженный (дискретное упорядочение $\mathrm{equ}^*$) и поэтому сохраняет все пределы, но не имеет правого сопряженного. Амальгамирование в общем случае превращает естественный частичный порядок

раздельного объединения $B \amalg (C \times W)$ в предпорядок, поэтому выполняется дополнительная факторизация по отношению, обозначаемому через $\overset{\leq}{>}^{\infty}$, которое отождествляет изоморфные элементы этого предпорядка (рассматриваемого как категория). Чтобы схема связывания порождала кодекартов квадрат на уровне аспектной структуры, связка должна быть совместима с конкуррентностью (concurrency) в том смысле, что она не должна фиксировать порядок вызова различных аспектов совета, привязываемых к одной точке соединения [20]. Это условие выполняется в технологиях АОП, позволяющих совету иметь только одну точку вызова, например в AspectJ. Формально оно задается следующим образом.

**Предложение 6.** Связывание пары **PosEqu**-морфизмов $j : B \leftarrow C \rightarrow W : e$ существует тогда и только тогда, когда для любых $x, y \in C$ условия $j(x) = j(y)$ и $x \leq y$ влекут $v \sim e(x)$ для любого $v \in W$ такого, что $e(x) \leq v \leq e(y)$.

Доказательство. Чтобы множество меток копредела схемы связывания было амальгамой ее str-образа, необходимо и достаточно, чтобы факторизация по отношению $\overset{\leq}{>}^{\infty}$ не привела к отождествлению различных меток. Это равносильно тому, что для любых $x, y \in C$ таких, что $j(x) = j(y)$ и $x \leq y$, отображение $\mathrm{str}(\langle 1_C, e\rangle) : \mathrm{str}(C) \rightarrow \mathrm{str}(C) \times \mathrm{str}(W)$ устанавливает биекцию между множествами $\mathrm{str}(\{z | x \leq z \leq y\})$ и $\mathrm{str}(\{(z, v) | x \leq z \leq y \wedge e(x) \leq v \leq e(y)\})$, т. е. тому, что $|\mathrm{str}(\{v | e(x) \leq v \leq e(y)\})| = 1$. □

Рассмотрим связывание сценариев при проектировании информационно-управляющих систем с применением событийной модели. Здесь совет представляет собой обработчик событий базы, образующих срез. Инструмент связывания можно рассматривать как монитор исполнения сценария базы, который при обнаружении точек соединения вызывает исполнение сценариев связываемых с ними советов. Сценарий функционирования монитора описывается связкой — центральным объектом в спецификации связывания.

Таким образом, событийное программирование (event-based programming) является одним из способов реализации аспектно-ориентированного подхода [20]. Оно придает системе способность реагировать на изменения в окружении, регистрируемые в виде событий, путем динамической подстройки хода исполнения процессов [26].

Крупноблочный базовый сценарий $B$ основного процесса функционирования системы состоит в повторяющемся исполнении следующей цепи классов задач по обработке событий, возникающих на объекте управления:

$$\text{регистрация} \rightarrow \text{сохранение} \rightarrow \rightarrow \text{анализ} \rightarrow \text{воздействие}\,.$$

Инфраструктурные аспекты, обеспечивающие его корректную работу, например паспорт объекта и средства защиты информации, комплексируются с ним путем связывания, нарушая возможность разделения сценария на модули. Чтобы исполнять разные стадии основного цикла обработки событий на разных аппаратных узлах, необходимо реплицировать инфраструктуру между ними. Такая репликация вызывает основные трудности при проектировании и эксплуатации распределенных систем по сравнению с локальными (автоматизированными системами управления технологическими процессами), поэтому привлечение технологий АОП способствует снижению затрат.

Рассмотрим следующий пример. При сохранении события в базу данных системы может потребоваться внести изменение в паспорт объекта, если оно сигнализирует о фактических изменениях его структуры, например о замене прибора. Другие фрагменты паспорта могут измениться по результатам анализа, если выявлено его несоответствие объекту: например, зарегистрировано потребление энергии единицей оборудования, не связанной в паспорте ни с каким питающим центром. Поведение аспекта изменения паспорта $W$ можно упрощенно описать сценарием из двух событий, относящихся к одному классу задач:

$$\text{изменение}_{\text{запрос}} \rightarrow \text{изменение}_{\text{внес}}\,.$$

Его связка с базовым сценарием имеет вид дискретного двухэлементного сценария $\mathbf{1} \amalg \mathbf{1}$ —, где $\mathbf{1}$ — терминальный **PosEqu**-объект: морфизм $j$ представляет собой его биекцию на подмножество $\{\text{сохранение}, \text{анализ}\} \subseteq B$, а $e$ — постоянное отображение на элемент $\text{изменение}_{\text{запрос}} \in W$. Как легко проверить, связывание существует и порождает следующий сценарий:

$$\begin{array}{ccccccc} \text{регистрация} \rightarrow & \text{сохранение} & \rightarrow & \text{анализ} & \rightarrow & \text{воздействие}, \\ & \downarrow & & \downarrow & & \\ & \text{сохранение}_{\text{внес}} & & \text{анализ}_{\text{внес}} & & \end{array}$$

где индексом снабжены события соответствующих классов задач, фиксирующие внесение изменений в паспорт объекта. В соответствии с утверждением (ii) предложения 5 метка аспекта изменения паспорта в этом сценарии отсутствует — она «сливается» с метками точек соединения. В то же время каждый акт изменения паспорта приобретает пометку инициировавшей его задачи, которую можно

сохранить вместе с изменяемыми данными (при условии, что хранилище данных паспорта спроектировано на базе технологии ADM). Таким образом, задача ведения паспорта рассеивается по процессам-источникам данных с обеспечением трассирования каждого элемента к задачам, вызвавшим его изменение. Это позволяет обеспечить практически полную актуальность и достоверность паспорта.

## 5 Заключение

В настоящей работе представлена теоретико-категорная семантическая модель АОП, позволяющая формализовать процедуру связывания аспектов и доказать ее главные свойства. Она отличается от подходов к формализации АОП, предложенных в литературе, поскольку они представлены в терминах частных формализмов и могут применяться только в рамках частных парадигм программирования. Модели АОП строились с привлечением почти всех формализмов теоретического программирования, таких как алгебра процессов [27], лямбда-исчисление [28], преобразования графов [29], проверка на моделях [30], языки описания архитектуры [23] и др. В противоположность им была предложена и теоретически обоснована единая концепция технологии АОП с приложением к моделированию данных и сценариев исполнения процессов — ключевых составляющих широкого класса программных систем. Эта концепция предлагает явно размечать модели массивов данных и процессов на фрагменты, образующие средства решения отдельных классов задач, и в дальнейшем оформлять их самостоятельными единицами комплексирования — аспектами. Вхождения аспектов удается формально идентифицировать (трассировать) по ходу сборки системы, что снижает затраты на ее создание и эксплуатацию.

Такой подход был апробирован при проектировании программной платформы учета и управления энергообеспечением «Энергиус» [31], на базе которой были созданы компоненты крупномасштабных систем диспетчерского управления [18], интеллектуального учета электроэнергии [32], управления энергоэффективностью [33] и др. В ней были реализованы изложенные в настоящей работе проектные решения в области аспектно-ориентированного моделирования данных [18] и процессов [20], что позволило снизить ее стоимость по сравнению с аналогичными программными продуктами. Конечно, предложенный подход нуждается в развитии и расширении сферы применения — здесь возникают задачи, представляющие перспективные направления дальнейших исследований.

## Литература


1. *Kiczales G., Lamping J., Mendhekar A., et al.* Aspect-oriented programming // Lecture Notes in Computer Sci., 1997. Vol. 1241. P. 220–242.
2. *Steimann F.* The paradoxical success of aspect-oriented programming // OOPSLA'06 Proceedings. — Portland, 2006. P. 481–497.
3. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами приложений на C++. — 2-е изд. — М.: Бином; СПб.: Невский диалект, 1999.
4. *Fiadeiro J. L.* Categories for software engineering. — Berlin–Heidelberg–N.Y.: Springer, 2005.
5. *Morin B., Barais O., Jézéquel J. M.* Weaving aspect configurations for managing system variability // 2nd Workshop (International) on Variability Modelling of Software-Intensive Systems VaMoS'08 Proceedings. — Essen, 2008. P. 53–62.
6. *Adams B., De Schutter K., Zaidman A., Demeyer S., Tromp H., De Meuter W.* Using aspect orientation in legacy environments for reverse engineering using dynamic analysis — an industrial experience report // J. Syst. Software, 2009. Vol. 82. No. 4. P. 668–684.
7. *Colyer A., Clement A., Harley G., Webster M.* Eclipse AspectJ. — Reading: Addison-Wesley, 2004.
8. *Rashid A., Chitchyan R.* Aspect-oriented requirements engineering: A roadmap // 13th Workshop (International) on Early Aspects EA'2008 Proceedings. — Leipzig, 2008. P. 35–41.
9. *Brichau J., Chitchyan R., Rashid A., D'Hondt T.* Aspect-oriented software development: An introduction // Wiley Encyclopedia of Computer Science and Engineering. Vol. 1. — N.Y.: Wiley and Sons, 2008. P. 188–198.
10. *Ожегов С. И., Шведова Н. Ю.* Толковый словарь русского языка. — М.: Азъ, 1992.
11. *Голдблатт Р.* Топосы. Категорный анализ логики. — М.: Мир, 1983.
12. *Adámek J., Herrlich H., Strecker G.* Abstract and concrete categories. — N.Y.: Wiley and Sons, 1990.
13. *Ковалёв С. П.* Применение аспектно-ориентированного подхода для автоматизации крупномасштабных объектов и процессов управления // Управление развитием крупномасштабных систем (MLSD'2012): Мат-лы VI Междунар. конф. — М.: ИПУ РАН, 2012. Т. 2. С. 315–318.
14. *Ковалёв С. П.* Формальный подход к аспектно-ориентированному моделированию сценариев // Сиб. журн. индустр. математики, 2010. Т. 13. № 3. С. 30–42.
15. *Kovalyov S. P.* Modeling aspects by category theory // 9th Workshop on Foundations of Aspect-Oriented Languages Proceedings. — Rennes, France, 2010. P. 63–68.
16. *Бениаминов Е. М.* Алгебраические методы в теории баз данных и представлении знаний. — М.: Научный мир, 2003.

17. *Hruška T., Kolenčík P.* Comparison of categorical foundations of object-oriented database model // Lecture Notes Computer Sci., 1997. Vol. 1341. P. 302–319.
18. *Андрюшкевич С. К.* Построение информационной модели крупномасштабных объектов технологического управления с применением аспектно-ориентированного подхода // Вестник НГУ. Сер. Информационные технологии, 2010. Т. 8. № 3. С. 34–45.
19. *Коберн А.* Современные методы описания функциональных требований. — М.: Лори, 2002.
20. *Андрюшкевич С. К., Ковалёв С. П.* Динамическое связывание аспектов в крупномасштабных системах технологического управления // Вычисл. технологии, 2011. Т. 16. № 6. С. 3–12.
21. *Pratt V. R.* Modeling concurrency with partial orders // Int. J. Parallel Programming, 1986. Vol. 15. No. 1. P. 33–71.
22. *Douence R., Fradet P., Südholt M.* Trace-based aspects // Aspect-Oriented Software Development. — Reading: Addison Wesley, 2004. P. 201–218.
23. *Pinto M., Fuentes L., Troya J. M.* DAOP-ADL: An architecture description language for dynamic component and aspect-based development // Lecture Notes Computer Sci., 2003. Vol. 2830. P. 118–137.
24. *Nakajima S., Tamai T.* Weaving in role-based aspect-oriented design models // Early Aspects'2004: Workshop Proceedings. — Vancouver, Canada, 2004. http://trese.cs.utwente.nl/workshops/oopsla-early-aspects-2004/Papers/NakajimaEtAl.pdf.
25. *Smith D. R.* Composition by colimit and formal software development // Lecture Notes Computer Sci., 2006. Vol. 4060. P. 317–332.
26. *Hermosillo G., Seinturier L., Duchien L.* Using complex event processing for dynamic business process adaptation // 7th IEEE Conference (International) on Services Computing SCC'2010 Proceedings. — Miami, 2010. P. 466–473.
27. *Andrews J. H.* Process-algebraic foundations of aspect-oriented programming // Lecture Notes in Computer Sci., 2001. Vol. 2192. P. 187–209.
28. *Jagadeesan R., Pitcher C., Riely J.* Open bisimulation for aspects // AOSD'07 Proceedings. — Vancouver, Canada, 2007. P. 107–120.
29. *Whittle J., Jayaraman P.* MATA: A tool for aspect-oriented modeling based on graph transformation // Lecture Notes Computer Sci., 2008. Vol. 5002. P. 16–27.
30. *Katz E., Katz S.* Verifying scenario-based aspect specifications // Lecture Notes Computer Sci., 2005. Vol. 3582. P. 432–447.
31. *Ковалёв С. П., Андрюшкевич С. К., Гуськов А. Е.* Интеграционная платформа учета и управления энергообеспечением «Энергиус»: Свидетельство о государственной регистрации программы для ЭВМ № 2009613359 от 26 июня 2009 г.
32. *Андрюшкевич С. К., Ковалёв С. П., Кубышкин А. С., Трегубов А. М.* Проблемы автоматизации управления процессами розничного рынка электроэнергии // Проблемы управления и моделирования в сложных системах (ПУМСС-2012): Труды XIV Междунар. конф. — Самара: СамНЦ РАН, 2012. С. 376–386.
33. *Ковалёв С. П., Пароджанов С. С.* Концепция создания автоматизированной системы мониторинга и управления энергоэффективностью на объектах города Москвы // Информационно-измерительные и управляющие системы, 2011. Т. 9. № 6. С. 50–58.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 81–93

# КОГНИТИВНАЯ ИНТЕРОПЕРАБЕЛЬНОСТЬ ЭКСПЕРТНОГО ВЗАИМОДЕЙСТВИЯ В ЗАДАЧЕ ОБРАБОТКИ РУССКО-ФРАНЦУЗСКИХ ПАРАЛЛЕЛЬНЫХ ТЕКСТОВ: ЛИНГВОКОГНИТИВНЫЕ АСПЕКТЫ

О. С. Кожунова[1]

**Аннотация:** Обсуждаются ресурсы информационно-коммуникационных технологий (ИКТ) «Пополняемая база лингвистических данных по трудностям перевода» и «Специальный тезаурус русско-французских параллельных текстов», которые находятся на стадии проектирования и будут разработаны одновременно с созданием параллельного корпуса русско-французских художественных текстов. Помимо их функциональности рассматриваются лингвокогнитивные аспекты взаимодействия экспертов различных областей, решающих задачу обработки русско-французских параллельных текстов совместными усилиями.

**Ключевые слова:** когнитивная интероперабельность; задача обработки естественного языка; русско-французские параллельные тексты

## 1 Введение

Сегодня задача автоматической обработки параллельных текстов и создания соответствующего инструментария в помощь филологу, который занимается сравнительным языкознанием, переводоведением и другими аспектами анализа переводных текстов, находится на пике своей актуальности. Это происходит, поскольку ИКТ достигли уровня развития, позволяющего моделировать и частично замещать деятельность экспертов. Особенно ресурсы ИКТ востребованы в задачах машинного перевода, сопоставления параллельных текстов на разных языках, сопоставления языковых структур различного уровня для проведения лингвистического анализа текстов, формирования выводов об интересующем исследователя языковом явлении, например о поведении грамматической конструкции, использования выразительных средств в языке и т. п.

За рубежом активные попытки создания необходимого инструментария предпринимались уже в 1990-х гг. и в настоящее время приобрели более сфокусированный характер, т. е. нацелены на решение специфических проблем и задач корпусных исследований [1–4].

Отечественные филологи и специалисты по компьютерной лингвистике также заинтересованы в решении узкоспециализированных задач корпусной лингвистики, но помимо этого предпринимают попытки к созданию универсальных ресурсов для обработки и выравнивания параллельных текстов [5, 6].

Обсуждаемые в настоящей работе ИКТ-ресурсы «Пополняемая база лингвистических данных по трудностям перевода» и «Специальный тезаурус русско-французских параллельных текстов» находятся на стадии проектирования и будут разработаны одновременно с созданием параллельного корпуса русско-французских художественных текстов. Эти ресурсы позволят не только выйти на качественно иной уровень работы с параллельными текстами, в том числе в составе лингвистического корпуса, но и решить те актуальные задачи, которые сегодня стоят перед филологами, работающими в областях сравнительного языкознания и переводоведения, а именно: выявление и фиксация трудностей русско-французского перевода, систематизация и типология таких трудностей с предложениями по их разрешению и примерами из параллельных текстов, составление тезауруса переводных русско-французских терминов определенной стилистической направленности, установление необходимых связей и отношений между ними и т. д.

Кроме научно-исследовательского и практического применения вышеупомянутых лингвистических ресурсов предполагается также использовать их для проведения дистанционных корпусных исследований студентами, аспирантами и докторантами и пополнить их для этих целей учебными

[1]Институт проблем информатики Российской академии наук, kozhunovka@mail.ru

программами выравнивания параллельных художественных текстов.

В работе уделяется особое внимание понятию когнитивной интероперабельности[1] экспертного взаимодействия в рамках вышеупомянутой задачи в силу ее междисциплинарности и сложности в установлении такого взаимодействия между экспертами из разных предметных областей.

## 2 Параллельный корпус и лингвистические ресурсы информационно-коммуникационных технологий

Параллельный корпус русско-французских художественных текстов необходим по многим причинам. Во-первых, в виду активного создания и редактирования «Национального корпуса русского языка» [7] соответствующие инициативы, связанные с созданием параллельных корпусов, являются ожидаемыми и востребованными, в том числе и в рамках международного научно-исследовательского сотрудничества в этой области. Во-вторых, на материале художественных текстов языковые несоответствия русского и французского языков и сложности перевода с одного языка на другой выступают наиболее ярко, поэтому предлагается в первую очередь создавать соответствующий корпус именно художественных параллельных текстов.

Построение пополняемой базы лингвистических данных по трудностям перевода является первым шагом в создании необходимых ИКТ-ресурсов, сопровождающих корпус параллельных русско-французских текстов. Эта база лингвистических данных предназначена для формализации типологии трудностей перевода, установления причинно-следственных и иных отношений между отдельными трудностями и целыми классами, сопоставления отдельных трудностей с примерами из текстов и даже для размещения вариантов разрешения обозначенных трудностей перевода. Такая база данных предоставит богатый материал не только для филологических исследований в области переводоведения и сравнительного языкознания, но и позволит сделать качественный рывок в практическом русско-французском переводе [8].

Формирование учебных программ выравнивания параллельных художественных текстов позволит вовлекать в решение важных проблем корпусной лингвистики, сравнительного языкознания и разделов филологии, связанных с переводоведением, студентов, аспирантов и молодых специалистов, развивать у них навыки практических научных работ во многих разделах филологии, а также позволит готовить новые кадры для будущих исследований в этой области.

Помимо вышеперечисленных ресурсов необходимо подчеркнуть важность создания специального тезауруса русско-французских параллельных текстов. Поскольку в рамках построения параллельных русско-французских корпусов возникает множество филологических и лингвистических задач, связанных как с извлечением данных и знаний из текста, сопоставлением структур разного уровня анализа, так и с представлением знаний, описанных в тексте, классификацией терминов и связей между ними, задача построения тезауруса русско-французских терминов является актуальной. Кроме того, решение этой задачи дополняет процесс создания параллельных корпусов и релевантные исследования. Аккумуляция русско-французской лексики определенной стилистики с одновременной фиксацией семантических, родовидовых и других тезаурусных отношений позволит качественно изменить подход к анализу русско-французских текстов и содержащихся в них языковых структур и отношений.

Идея построения специального тезауруса и его применения в задаче обработки параллельных русско-французских текстов с последующим формированием соответствующего корпуса основана на успешном опыте масштабных проектов по созданию многоязычных тезаурусов и тезаурусоподобных лингвистических ресурсов. Одним из наиболее распространенных типов таких ресурсов являются автоматизированные словари, построенные по модели WordNet [9–12].

Проект по разработке словаря Princeton WordNet (PWN) английского языка в Принстонском университете (США) стартовал в первой половине 1980-х гг. и продолжается по сей день. Сейчас уже доступна версия 2.0 WordNet. Существующая версия охватывает более 120 тыс. слов общеупотребительной лексики современного английского языка [13].

Этот словарь — базу данных тезаурусного типа — можно использовать для различных лингвистических задач. В частности, при проведении информационного поиска wordnet-словари применяются для расширения запроса пользователя за счет парадигматически и синтагматически связанных слов, например компонентов синсета (множества синонимов, объединенных в набор) вместе с

[1] Интероперабельность (*англ.* interoperability) — способность к взаимодействию.

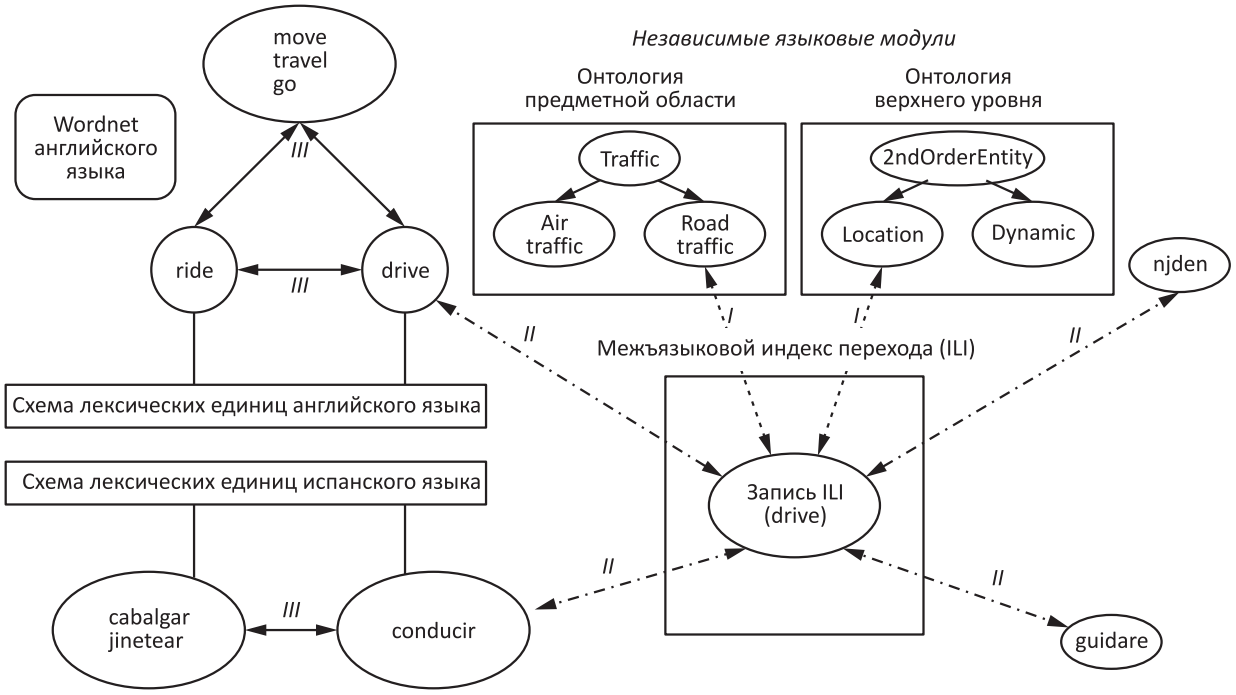



**Рис. 1** Фрагмент архитектуры базы данных EuroWordNet для английского и испанского языков

его гипонимами и согипонимами или связей типа «глагол–актант», которые дают возможность осуществлять контекстный поиск. Данные о синтагматических отношениях слов позволяют применять wordnet-словари для решения задачи снятия неоднозначности смысла слова. Wordnet можно использовать для вычисления смысловой близости текстов на основе гиперонимических отношений. Wordnet-словари могут служить лексиконом для формальных грамматик. Формат wordnet является удобным формализмом для представления состава и структуры лексики специальных подъязыков (например, медицинских, экономических терминов). Wordnet-словари являются удобным инструментом для проведения исследований в области лексической семантики, например гипонимические отношения в wordnet-словарях позволяют определять направление метонимических переносов и прогнозировать появление новых лексико-семантических вариантов [14].

За период с марта 1996 г. по сентябрь 1999 г. при финансировании Европейской комиссии был создан многоязычный вариант WordNet — EuroWordNet [15], что стало новым этапом в эволюции wordnet-словарей. В рамках европейского проекта было создано не только несколько тезаурусов для европейских языков (голландского, испанского, итальянского, немецкого, французского, чешского и эстонского), но и впервые была реализована идея объединения отдельных wordnet-представлений в общую систему. Все компоненты EuroWordNet были построены по единой модели, что, однако, не предполагало прямого перевода английского варианта WordNet 1.5. Перед разработчиками стояла задача — отразить все особенности лексических систем национальных языков.

Совместимость компонентов EuroWordNet была обеспечена единством принципов и заданным набором общих понятий (Basic Concepts), на основе которых определялась система межъязыковых отсылок (Inter-Lingual-Index), дающих возможность переходить от лексикализованных значений одного языка к сходным, но не обязательно тождественным значениям в другом языке. Данный индекс позволяет использовать EuroWordNet не только для информационного поиска в рамках одного языка, но и для многоязычного поиска (рис. 1).

В рамках проекта EuroWordNet первоначальная структура словаря претерпела серьезные изменения. Был расширен набор семантических отношений за счет парадигматических отношений, связывающих слова разных частей речи (например, XPOS_NEAR_SYNONYMY: dead–death; XPOS_HYPERONYMY: to love – emotion; XPOS_ANTONYMY: to live – dead) и синтагматических отношений между глаголами и актантами-существительными (например, ROLE_INSTRUMENT: to write – pencil). Был сфор-

мирован новый подход к построению wordnet-словарей: с опорой на использование лексикографических источников (толковых, переводных и синонимических словарей) и результатов обработки корпусов современных текстов.

Успешное завершение проекта EuroWordNet послужило толчком к созданию большого числа wordnet-представлений для языков разных типов (например, венгерского, турецкого, арабского, тамильского, китайского и пр.), а также многоязычных ресурсов типа EuroWordNet (например, проект BalkaNet нацелен на объединение греческого, румынского, болгарского, сербского, турецкого и чешского wordnet-словарей). В 2001 г. была создана Всемирная Ассоциация WordNet (Global WordNet Association), целью которой является объединение уже существующих и только развивающихся национальных ресурсов этого типа, усовершенствование системы межъязыковых индексов и разработка общих стандартов, позволяющих использовать модель WordNet для языков разных типов [5].

С 1999 г. на кафедре математической лингвистики СПбГУ исследовательская группа под руководством И. В. Азаровой (Азарова И. В., Митрофанова О. А., Синопальникова А. А. и др.) ведет работы по проекту RussNet — созданию русской версии компьютерного словаря типа WordNet [16]. В задачи проекта входит построение лексико-семантического ресурса для отражения организации лексической системы русского языка в целом, для представления ядра его общеупотребительной лексики и фиксации семантических, семантико-грамматических и семантико-деривационных отношений русского языка. Кроме того, в настоящее время в Петербургском государственном университете путей сообщения разрабатывается проект русской версии WordNet под руководством С. А. Яблонского и А. М. Сухоногова [13]. Поэтому предполагается, что построение специального тезауруса русско-французских параллельных текстов с учетом международного опыта формирования многоязычных тезаурусов и особенностей междисциплинарной задачи создания и обработки корпуса параллельных текстов позволит проводить филологические исследования большего масштаба и глубины.

Поскольку сама задача создания корпуса параллельных текстов является междисциплинарной, то методы и подходы, задействованные в ее решении, так же разнообразны, а именно: методы системного анализа, искусственного интеллекта, компьютерной лингвистики, психолингвистики, когнитивного моделирования и т. п. В частности, используются методы извлечения данных из текстов, подходы к представлению знаний, классификации терминов и методы построения и обработки запросов на поиск в слабоструктурированных полнотекстовых документах, методы моделирования человеческого восприятия информации различного уровня сложности и т. п. Тем самым предполагается вовлечение в работу экспертов из нескольких областей: компьютерной лингвистики, текстологии, переводоведения, искусственного интеллекта, когнитивной науки, психологии и др. Такое многообразие специалистов, безусловно, повлечет за собой ряд затруднений в решении поставленной задачи, поскольку их картины мира, акценты и ракурсы взгляда на одни и те же данные / ситуативные контексты / проблемы изначально отличаются друг от друга. Предложенные способы согласования восприятия информации и соответствующих концептуальных связях будут описаны в разд. 3.

## 3 Опыт междисциплинарных задач

Ранее было сказано о междисциплинарности задачи построения русско-французского корпуса и соответствующих лингвистических ресурсов. Действительно, для решения такой многоплановой задачи необходимо обратиться к экспертным знаниям и компетенциям из разных областей. Однако и сама идея вовлечения экспертов с разными подходами к мировосприятию и научно-исследовательскими парадигмами таит в себе еще одну научную проблему: обеспечение взаимодействия экспертов из разных областей для выполнения одной задачи и согласование их картин мира, образов ситуации, ракурсов и подходов к решению возникающих вопросов.

Несмотря на новизну этой проблемы, у автора уже имеется опыт работы с междисциплинарными задачами и подходы к согласованию действий и понятий экспертов в разных областях знаний. В первую очередь, опыт получен в рамках подготовки диссертационной работы [17], посвященной созданию и исследованию технологии разработки семантического словаря показателей для систем информационного мониторинга. Одной из проблем, решаемой в рамках этой работы, была проблема различия в понимании экспертами смысла индикаторов, поскольку это является серьезным препятствием в реализации всех трех основных процедур, необходимых для оценивания программной деятельности в сфере науки: информационный мониторинг, анализ, получение количественных и экспертных оценок ее результатов, эффективности и результативности. Это вызвало необходимость решения задачи согласования понимания индикато-

ров разными экспертами. Было отмечено, что в силу особенностей формирования терминов мониторинга возникает также задача частной референции, когда одно название индикатора может обозначать целый класс индикаторов (например, индексы цитирования, смысл которых зависит от учета самоцитирования, а также цитирования соавторами и т. п.). Поэтому был предложен и разработан семантический словарь, который позволил отобразить информацию и связи, необходимые для решения задачи. Новизна предложенного семантического словаря состоит в том, что он содержит ссылки на алгоритмические и информационные ресурсы системы информационного мониторинга, а также нормативные документы как источники терминов рассматриваемой предметной области. Построенная формализация процесса извлечения терминов мониторинга и их определений из массива текстов (нормативные документы, научные статьи и т. д.) позволила выделить необходимые индикаторы, связи между ними и их определения. Это облегчило дальнейшую интеграцию индикаторов и других показателей мониторинга в классификационную схему семантического словаря (рис. 2 и 3) [17].

После завершения диссертационного исследования была начата научно-исследовательская работа «Лексико-семантические методы создания проблемно-ориентированных лингвистических ресурсов информационных систем», выполняемая в Институте проблем информатики РАН в 2011–2013 гг. Само название и поставленные задачи уже говорят о междисциплинарности этого исследования:

(1) семантическое моделирование предметных областей в рамках научно-исследовательской работы (НИР) как основа создания лингвистического обеспечения информационных систем;

(2) модификация лингвистических ресурсов систем информационного мониторинга с применением лексико-семантических методов в процессе создания систем информационного мониторинга научной деятельности;

(3) формализация и компьютерное моделирование трудностей перевода.

Из перечисленных задач реализованы и успешно сданы подзадачи задач 1–2, задача 3 находится в процессе выполнения. В ходе выполнения НИР был получен опыт и разработаны подходы к извлечению знаний из слабоструктурированных текстов, решению задач компьютерной лингвистики применительно к области информационного мониторинга и их формализации и моделированию в рамках разрабатываемой информационной системы, проектированию и созданию макета лингвистических ресурсов информационной системы (семантический и проективный словари), компонентному анализу лексических единиц на немецком языке и установлению лексико-смысловых связей с семантическими единицами русского языка (на примере русско-немецких языковых пар патентных текстов), выявлению различных языковых трансформаций и их классификации, а также разработка метапредставлений для компонентного лексико-семантического анализа [18–21].

Разработанные в рамках НИР подходы были успешно применены в междисциплинарных зада-

Показатели
1. Индикаторы
   1.1 Индикаторы результатов фундаментальных научных исследований (научные результаты)
      1.1.1 Непосредственные результаты
      1.1.2 Целевые результаты
      1.1.3 Индикаторы взаимосвязей и влияния научных результатов
         1.1.3.1 Взаимосвязи и влияние на здравоохранение
         1.1.3.2 Взаимосвязи и влияние на развитие сферы науки
         1.1.3.3 Взаимосвязи и влияние на развитие технологий
            1.1.3.3.1. Индексы самоцитирования в патентах
         1.1.3.4 Взаимосвязи и влияние на образование
2. Критерии
3. Параметры
4. Экспертные оценки

**Рис. 2** Классификационная схема, использованная для построения структуры семантического словаря

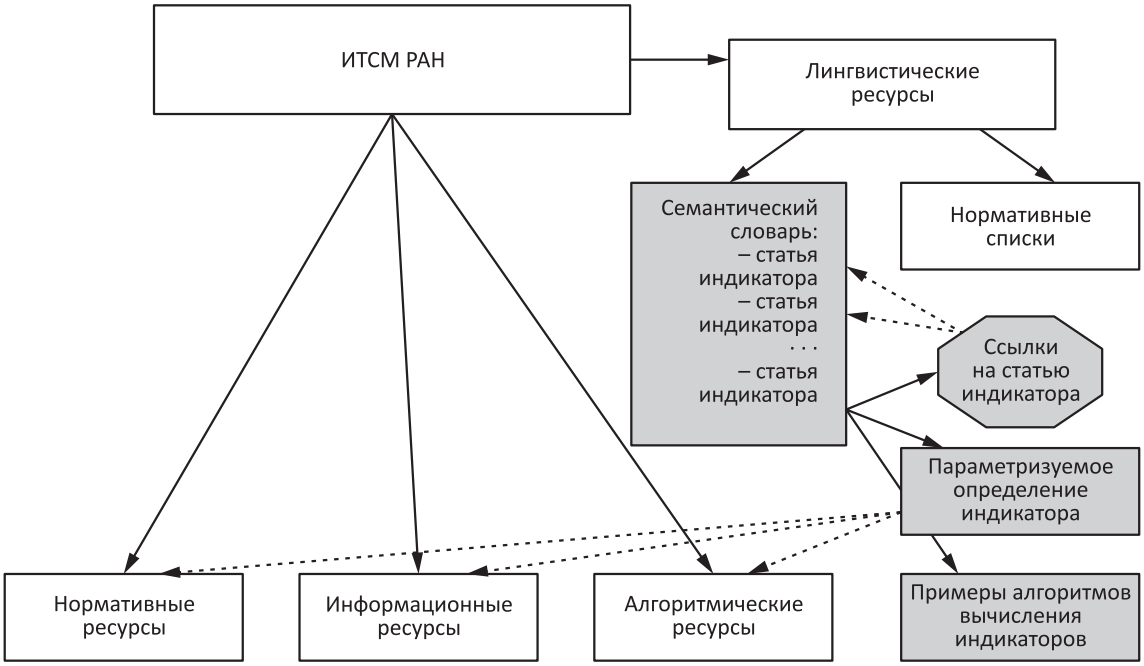



**Рис. 3** Связь статей семантического словаря с ресурсами ИТСМ РАН (ИТСМ — Информационно-технологическая система мониторинга, разрабатываемая в в отделе 16 ИПИ РАН)

чах информационного мониторинга и сопоставления параллельных патентных текстов [22, 23]. Поэтому при взаимодействии с экспертами геоинформатики, задачи которой также подразумевают вовлечение специалистов из нескольких предметных областей, были применены разработанные подходы, дополненные возможностями понятий когнитивной интероперабельности, когнитивного пространства и ряда междисциплинарных подходов и методов, образованных на стыке прикладной лингвистики, когнитивной психологии и искусственного интеллекта. Предлагаемый подход был успешно применен в междисциплинарных задачах геоинформатики [24].

В рамках данного подхода были предложены и частично реализованы следующие задачи:

- разработка лингвистических методов и моделей обеспечения когнитивной интероперабельности экспертной информационно-аналитической деятельности на основе геоинформационных описаний;
- формирование корпуса параллельных тематических текстов различных предметных областей на нескольких языках. Параллельные многоязычные корпуса в данном случае предназначены для накопления эмпирических данных по проблематике предметных областей, в частности в вопросах полноты и согласованности терминосистем, экспертного информационно-аналитического взаимодействия и т. д., а также для апробации созданного корпуса и когнитивно-лингвистических методов на массиве реальных кросс-языковых данных;
- разработка программных приложений для апробации предлагаемого подхода к моделированию когнитивного пространства взаимодействия экспертов в рамках заданной предметной области. Предполагается также апробация приложений в междисциплинарных задачах с целью разрешения терминологических разногласий между экспертами смежных областей, восстановления адекватных причинно-следственных связей между понятиями предметной области, обеспечения возможности принятия решений в случаях недоопределенных терминосистем или их отсутствия и других задачах;
- создание информационных технологий (ИТ) нового поколения, учитывающих особенности взаимодействия экспертов разного уровня как внутри предметных областей, так и в междисциплинарных задачах.

Предлагаемый подход был апробирован на практике в ходе координации экспертов-аналитиков, принимающих согласованные решения на основе анализа геоинформационных описаний (рис. 4).

Существующие методы лингвистического анализа, применяемые к задачам формирования и

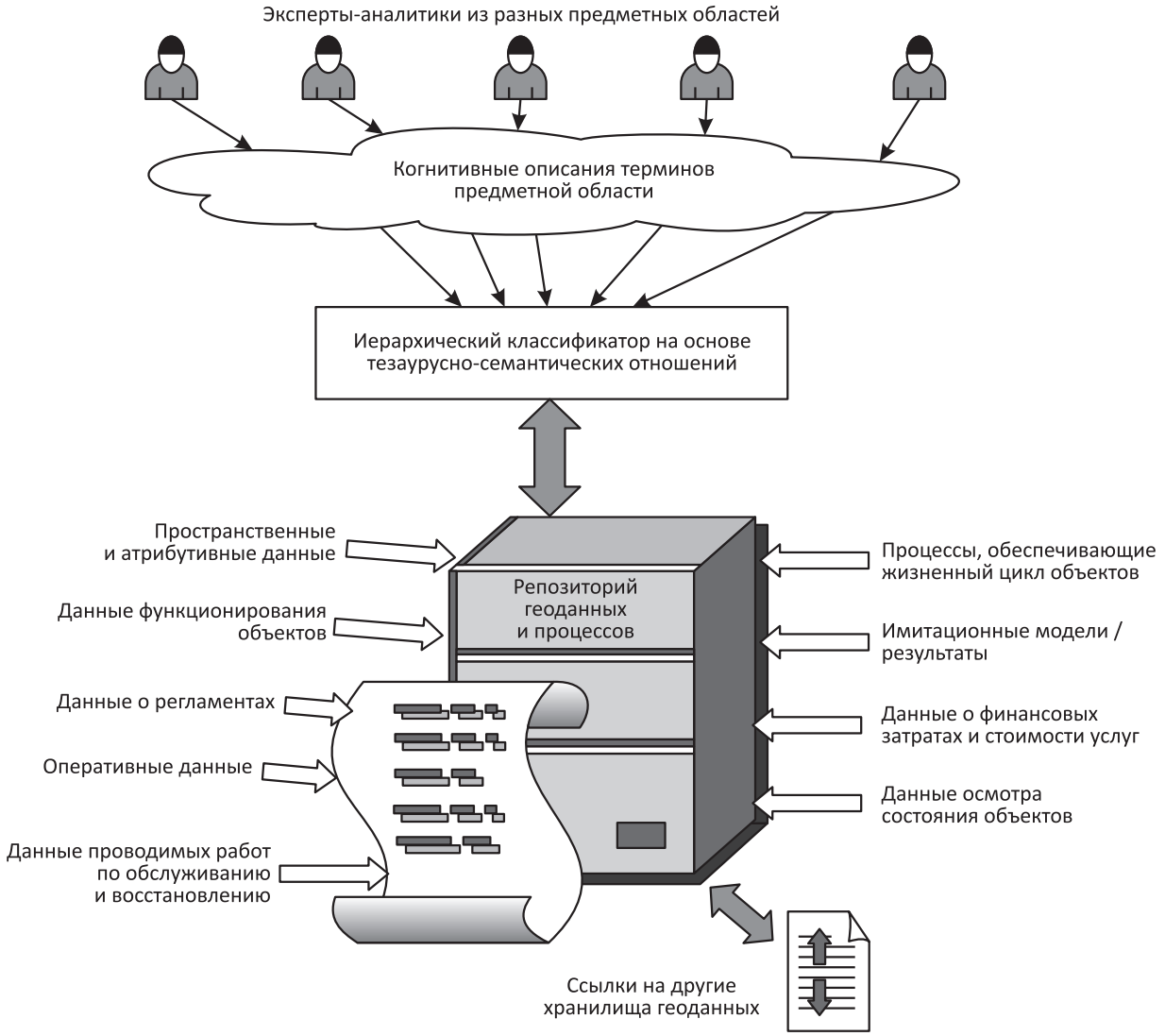



**Рис. 4** Взаимодействие экспертов из разных предметных областей в геоинформационной системе

сопровождения интеллектуальных геоинформационных систем, потребовали привлечения методов, которые позволили выделить необходимые единицы в соответствующих информационных структурах, фиксировать связи между ними, а также выявить и отобразить их семантику и ситуативный контекст [25].

Среди исследованных методов были выделены методы лексико-семантического моделирования когнитивных структур знаний, которые позволяют учесть особенности геоинформационной среды [24]. В работе [24], в частности, говорилось о том, что структура интеллектуального анализа геоинформационных данных основана на гибридных формах представления знаний, а именно иерархической классификации с возможностью установления и корректировки нескольких видов тезаурусно-семантических отношений (собственно семантические, отношения часть–целое, гипоним–гипероним и т. д.). В данной структуре, в отличие от онтологий, акцент смещен в сторону когнитивных аспектов описания и семантического наполнения терминов, которыми оперируют в работе эксперты различных областей и уровней компетенции, а не на описание самой предметной области.

Структура была спроектирована и представлена на основе xsd-схем, заполнение которых объектами отрасли и их связями было реализовано в отдельных xml-файлах. Анализ и валидация предложенной структуры осуществлены на основе программных модулей на C++. Затем был разработан подход на основе лексико-семантического моделирования к интеграции лингвистического обеспечения тезаурусного типа, совместимого с заданной структурой,

определенными в ней объектами и отношениями. В результате были получены программные модули на C++, разделенные по функциональности: поддержка структуры интеллектуального анализа данных в системе, сопровождение и наполнение тезауруса геоинформационной системы, обеспечение согласованной интеграции тезауруса и сопряженного лингвистического обеспечения в геоинформационную систему [24].

Следующим ключевым понятием, задействованным в разработке междисциплинарного подхода к координации экспертов-аналитиков, принимающих согласованные решения на основе анализа геоинформационных описаний, было понятие когнитивного пространства. Когнитивное пространство с учетом многофакторности взаимодействия экспертов на основе геоинформационных описаний расширяет понятие единого геоинформационного пространства [26]. Были сформулированы требования к разрабатываемой архитектуре единого геоинформационного пространства на основе сравнительного анализа существующих метасхем баз геоданных с учетом семантики, заложенной в пространственные онтологии. Проведенный анализ исследований показал необходимость разработки модели, учитывающей латентные атрибутивные связи между онтологиями разнородных геоданных, что должно повысить качество отображаемых связей в концептуальной схеме базы геоданных. Предлагаемая архитектура представляет собой многоуровневую структуру базы геоданных, при этом каждый из уровней ориентирован на решение определенного класса задач и содержит определенный набор пользовательских и библиотечных процедур управления процессами обработки геоданных для создания инструментария управления программным комплексом [24].

Следует отметить, что впервые инструментарий и возможности лексико-семантических методов и моделирования были применены в задачах геоинформатики. Интеграция лингвистического обеспечения тезаурусного типа на основе лексико-семантического моделирования позволила сгруппировать разнородные геоданные и структуры в едином геоинформационном пространстве в рамках разработанной гибридной формы представления знаний — иерархической классификации с элементами лексико-семантических отношений, а также выявить объекты и понятия данной предметной области и фиксировать тезаурусно-семантические отношения между ними [24].

Методы и подходы лексико-семантического моделирования, которые в настоящее время широко применяются в различных предметных областях, впервые были использованы при проектировании архитектуры лингвистического обеспечения геоинформационного пространства, формировании структуры интеллектуального анализа геоданных в геоинформационных системах и интеграции лингвистического обеспечения тезаурусного типа в единое геоинформационное пространство. Лексико-семантическое моделирование применительно к задачам и проблемам геоинформатики позволяет использовать средства анализа глубинных структур языка для извлечения геоданных, их связей и отношений, встраивания их в заданную геоинформационную структуру и верификации их семантики в разрезе формирования единого геоинформационного пространства. Инструменты глубинного представления языковых структур способствуют разрешению многочисленных неоднозначностей геоданных и связанной с этим рассогласованности при принятии важных решений экспертами-аналитиками, а также формировать единое геоинформационное пространство, позволяющее отображать адекватную информационную структуру геоданных [24].

В качестве возможных приложений предложенного подхода рассматриваются следующие задачи: координация работы экспертов из различных предметных областей, в том числе в междисциплинарных задачах, в информационных системах в автоматическом/полуавтоматическом режиме; междисциплинарные научные исследования различных предметных областей с целью разработки проблемно-ориентированных программных приложений и установления взаимодействия экспертов различного профиля и уровня подготовки; изучение методов и подходов компьютерной лингвистики и их адаптивных приложений для создания нового поколения ИТ.

## 4 Когнитивная интероперабельность и предлагаемый подход

Вышеописанная структура представления знаний хорошо себя показала применительно к задачам информационного мониторинга и геоинформатики. Но будет ли она так же универсальна, если эксперты столкнутся с ситуациями неопределенности ключевых понятий, с нечеткими постановками задач, с лексической полисемией и другими препятствиями к однозначной и эксплицитной трактовке терминов, их смысловых связей и их восприятия специалистами из разных областей (например, психологии, лингвистики, информатики и др.)?

Определенно, необходимо усовершенствовать предложенный подход с учетом указанных когнитивно-лингвистических механизмов восприятия информации человеком. По этому поводу было подготовлено много содержательных работ. Так, в [27] рассматриваются механизмы образования лексической полисемии и даже предлагается концептуально-смысловая модель ее образования. Моделирование взаимодействия смысловых значений слова и изучение истоков его многозначности существенно расширит возможности любого тезауруса предметной области. В частности, использование автором работы [27] понятия базового концепта [28] и дихотомии «основное vs. производное значение слова» [29] позволит моделировать потенциальные девиации смыслов терминов и позиционировать новые смыслы среди традиционных, а также выделять основные, «ядерные» понятия предметной области и выстраивать вокруг них такую иерархически-сетевую структуру представления знаний, которая наиболее полно отражает терминологический портрет изучаемой области/задачи/проблемы. Последние соображения связаны с понятием базового концепта, который определяется как обобщенный (концептуальный) объект, представляющий собой сложную когнитивную единицу — совокупность Формы, Действий и Интенций [27]:

$$\text{Концепт} = \text{концептуальный Объект} = \\ = (\text{Форма, Действия, Интенции}).$$

Под Формой здесь понимается структура элементарных пространственных объемов, под Интенциями — содержательные характеристики Формы (желания, цели, намерения, потребности, функции и т. п.), а Действия представляют собой типичные физические действия Формы, посредством которых реализуются ее Интенции [27]. Интересно, что такой концептуальный Объект задает свою категорию конкретных объектов, схожих с ним по всем трем характеристикам. В процессе когнитивного развития ребенка именно эта пара — концептуальный Объект и задаваемая им категория — позволяет познавать мир и расширять границы уже познанного. Это означает что, поскольку такое представление понятий не зависит от родного языка носителя и хранится в его долговременной памяти, механизм фиксации концептуальных Объектов и их категорий чрезвычайно важен при отображении знаний экспертов о какой-либо области и позволяет выработать обобщенный подход к формализации понятий и их связей. Что касается основных и производных значений [29], то, как было отмечено выше, их фиксация и встраивание в иерархически-сетевую структуру знаний некоторой области позволит не только моделировать потенциальные девиации смыслов терминов, но и позиционировать новые смыслы среди традиционных.

Такой подход к моделированию смыслов играет важную роль при взаимодействии экспертов из разных областей, так как производные смыслы не хранятся в памяти носителя языка, а основные смыслы одних и тех же понятий у разных людей могут существенно отличаться друг от друга (и даже у одних и тех же людей, но в разных языковых ситуациях) (см. пример 1). Это позволит обеспечить когнитивную интероперабельность [30] в их работе и достичь поставленных целей при решении ими междисциплинарных задач/проблем.

**Пример 1.**

*По полю рыжей стрелой летела лиса* (основное значение).

*Ну и лиса эта Ваша Маша!* (производное значение).

*Вон смотри, какая обезьяна!* (в зоопарке, про животное — основное значение).

*Фу, какая обезьяна!* (про уродливого человека или волосатого мужчину).

Приведенные выше соображения находят подтверждения и в других публикациях, в том числе в [31]. Авторы провели серию когнитивно-лингвистических экспериментов с носителями языка, что позволило им выявить интересные закономерности восприятия людьми лексических единиц языка и их смысловых связей. Например, при прослушивании устной речи неоднозначными оказываются единицы, совпадающие только по звучанию, но различающиеся написанием (омофоны: плот–плод, порог–порок и т. д.), а при восприятии письменной речи, напротив, — совпадающие по написанию, но различающиеся звучанием (омографы: зам**о**к — з**а**мок, м**у**ка — мук**а** и т. п.) [31]. Кроме того, Черниговская и соавторы подчеркивают, что выбор значения слова в первую очередь происходит на основании его структуры, а затем уже учитывается его контекст. Любопытно, что в одном из экспериментов, описанном ими, оценивалась роль частотности при интерпретации лексически неоднозначного фрагмента речи при восприятии омофонов. В результате была подтверждена первостепенная роль частотности словоформ при осуществлении выбора между омофонами.

Отсюда можно сделать вывод о том, что для успешного взаимодействия экспертов в рамках такой междисциплинарной задачи, как построение и ведение базы данных трудностей русско-французского перевода и обеспечения когнитивной интероперабельности их деятельности (т. е. в двух различных системах эксперты видят согласованные образы представляемой информации), дей-

ствительно необходим такой ресурс, как специальный тезаурус русско-французских параллельных текстов. Этот тезаурус позволит не просто зафиксировать специальные термины и понятия из различных областей и семантические связи между ними, но и отобразит информацию об особенностях восприятия этих терминов разными специалистами в разных речевых ситуациях и контекстах.

Такого рода информация справедливо отнесена к области «когнитивной семантики» автором работы [32], который, как и автор [27], апеллирует к подходам Д. Лакоффа [33]. Кузнецов подчеркивает, что значения слов возникают раньше, чем концептуальные структуры (из доконцептуального телесного опыта) [31]. Под доконцептуальными структурами здесь подразумеваются гештальты (единые ментальные образы) и образно-схематические структуры: вместилище, верх–низ, часть–целое и т. д. Причем связанные с ними концепты непосредственно значимы, что влияет на непосредственное, однозначное восприятие предложения или фразы. Поэтому понимание, согласно Лакоффу, есть не что иное, как способность соотносить концепты со своим опытом, включая доконцептуальный. Следовательно, чтобы обеспечить возможность понимания экспертами необходимых терминов/концептов/ситуаций и прочего в едином ключе, необходимо прежде всего облегчить восприятие ими заданной информации на максимально доступном уровне с точки зрения общечеловеческих когнитивных способностей такого рода. В частности, использовать образно-схематические структуры, которые дают возможность людям рассуждать быстрее машин [32].

За понятием образно-схематической структуры скрывается идея о том, что существуют определенные схемы, которые человек изначально накладывает на воспринимаемый мир. Например, людям свойственно представлять большие фрагменты своего повседневного опыта в терминах вместилища, в английском языке характеризуемого в первую очередь через предлоги «in» и «out». Так, мы выходим из (out) полусонного состояния, забытья, смотрим в (in) зеркало и т. п. Лакофф провел специальное лингвистическое исследование, в котором на материале 600 (!) глаголов английского языка демонстрируется категоризация по схеме «вместилище» [34].

Таким образом, адекватная с точки зрения человеческого восприятия форма представления знаний в специальном тезаурусе, в том числе доконцептуальных структур, может существенно облегчить восприятие информации, необходимой для взаимодействия экспертов, причем независимо от их изначальной профессиональной специализации.

**Пример 2.**

Фрагмент из повести А. П. Чехова «Скучная история» и ее художественный перевод на французский язык:

*В детстве и в юности я почему-то питал страх к швейцарам и к театральным капельдинерам. . .*

*Dans mon enfance et mon adolescence, j'avais, je ne sais pourquoi, peur* ***des Suisses*** *et des ouvreurs de theater. . .*

Здесь ***des Suisses*** дословно переводится с французского языка как «швейцарцы» (национальность).

Возможная информация в специальном тезаурусе по поводу этого случая (пример 2):

> Образно-схематическая структура повестей А. П. Чехова «Прислуга — швейцары, капельдинеры, горничные» сопоставляется с аналогичной французской структурой «portiers, concierges, ouvreurs de loges, femmes de chambre». В базе данных трудностей перевода этот случай фиксируется с отсылкой к тезаурусу для согласованного восприятия и отображения экспертами параллельных текстов с пометой возможной подмены французскими лингвистами термина «швейцар» термином «швейцарец».

Получившаяся концептуальная схема специального тезауруса русско-французских параллельных текстов, разработанная с учетом вышеизложенных дополнений по усовершенствованию подхода к обеспечению когнитивной интероперабельности

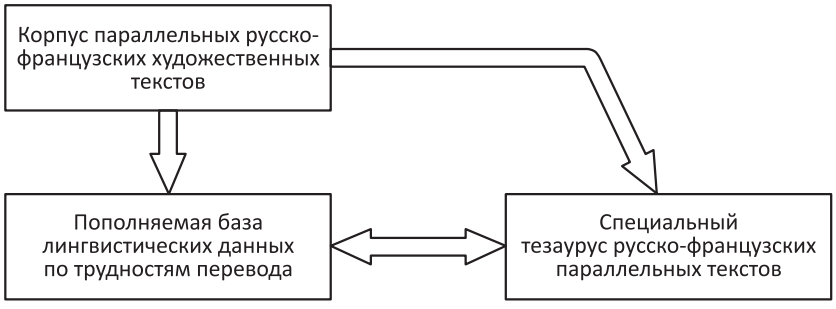



**Рис. 5** Специальный тезаурус русско-французских параллельных текстов и его связи с другими лингвистическими ресурсами

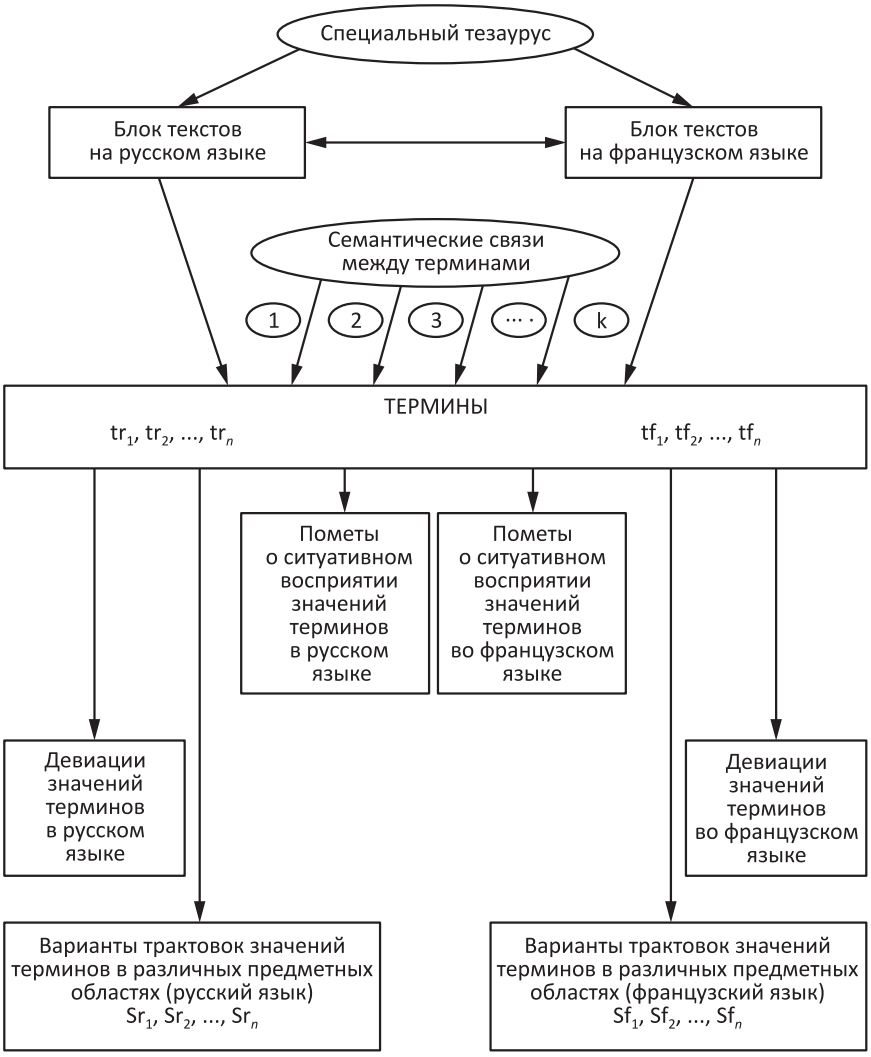



**Рис. 6** Концептуальная схема усовершенствованного специального тезауруса

экспертов различных областей, которые работают над созданием и обработкой параллельного корпуса русско-французских художественных текстов, представлена на рис. 5 и 6.

## 5 Заключение

Создание таких актуальных лингвистических ресурсов, как корпус русско-французских параллельных художественных текстов, пополняемая база лингвистических данных по трудностям перевода, учебные программы выравнивания параллельных художественных текстов, а также специальный тезаурус русско-французских параллельных текстов, влечет за собой необходимость совмещения разнообразных междисциплинарных подходов и, как следствие этого процесса, вызывает потребность в установлении взаимодействия и согласования базовых понятий у экспертов из разных областей.

В данной работе был приведен опыт работы с междисциплинарными задачами и краткое описание сформированных в результате подходов.

Далее были рассмотрены лингвистические аспекты когнитивного восприятия информации экспертами разных областей, включая механизмы выделения базовых концептов, дихотомию «основные vs. производные значения», фиксацию неоднозначных лексических единиц и ситуаций (омографы, омофоны, лексическая полисемия), учет частотности словоформ и ситуативного контекста в трактовке смысла терминов, отображение образно-схематических структур для определенных языковых ситуаций и т. д.

Рассмотренные аспекты позволили принять во внимание соответствующие особенности когнитивно-лингвистических механизмов восприятия информации и картины мира людьми — экспертами, а также усовершенствовать подход к обеспечению когнитивной интероперабельности экспертной деятельности на примере междисциплинарной задачи по созданию и обработке корпуса русско-французских параллельных художественных текстов и сопутствующих ресурсов (пополняемая база лингвистических данных по трудностям перевода; учебные программы выравнивания параллельных художественных текстов; специальный тезаурус русско-французских параллельных текстов). Все это позволило улучшить концептуальную схему специального тезауруса, тем самым подготовив почву к его разработке и апробации.

## Литература


1. *Sinclair J.* The automatic analysis of corpora // Directions in Corpus Linguistics: Nobel Symposium 82 Proceedings. — Berlin: Mouton de Gruyter, 1992.
2. *Wallis S., Nelson G.* Knowledge discovery in grammatically analysed corpora // Data Mining and Knowledge Discovery, 2001. Vol. 5. P. 307–340.
3. *Dukes K., Atwell E., Habash N.* Supervised collaboration for syntactic annotation of quranic arabic // Language Resources and Evaluation J., Special Issue on Collaboratively Constructed Language Resources, 2011.
4. *McCarthy D.* Exploiting distributional similarity for lexical acquisition // Компьютерная лингвистика и интеллектуальные технологии: По мат-лам ежегодной международной конф. «Диалог'2011». — М.: РГГУ, 2011. Вып. 10(17). С. 19–31.
5. *Азарова И. В., Синопальникова А. А., Яворская М. В.* Принципы построения wordnet-тезауруса RussNet // Компьютерная лингвистика и интеллектуальные технологии: Труды Междунар. конф. Диалог'2004. — М., 2004. С. 542–547.
6. *Ляшевская О. Н., Кузнецова Ю. Л.* Русский Фреймнет: к задаче создания корпусного словаря конструкций // Компьютерная лингвистика и интеллектуальные технологии: По мат-лам ежегодной Междунар. конф. «Диалог'2009». — М.: РГГУ, 2009. Вып. 8(15). С. 306–312.
7. Национальный корпус русского языка: Сайт проекта http://www.ruscorpora.ru.
8. *Бунтман Н. В., Зацман И. М.* О проекте создания компьютерного ресурса трудностей перевода: заметки на полях // Маргиналии-2010: границы культуры и текста: Тезисы II Междунар. конф. — М.: МГУ, 2010. С. 41–43. http://uni-persona.srcc.msu.su/site/conf/marginalii-2010/thesis.htm.
9. *Miller G. A.* Five papers on WordNet. CSL-Report. — Princeton: Princeton University, 1990. Vol. 43.
10. *Fellbaum C.* WordNet: An electronic lexical database. — Cambridge, 1998.
11. *Кожунова О. С.* Семантический словарь системы информационного мониторинга в сфере науки и ресурс Eurowordnet: структура, задачи и функции // Системы и средства информатики. — М.: Наука, 2008. Вып. 18. С. 156–170.
12. *Кожунова О. С.* Подходы к лексико-семантическому моделированию и лингвистические ресурсы информационных систем // Системы и средства информатики. — М: ИПИ РАН, 2011. С. 139–161.
13. *Сухоногов А. М., Яблонский С. А.* Словари типа WordNet в технологиях Semantic Web // Конф. по искусственному интеллекту (КИИ-2004): Тр. 9-й Национальной конф. по искусственному интеллекту с международным участием. — В 3-х т. — М.: Физматлит, 2004. Т. 2. С. 557–564.
14. *Азарова И. В., Митрофанова О. А., Синопальникова А. А., Ушакова А. А., Яворская М. В.* Разработка компьютерного тезауруса русского языка типа WordNet // Корпусная лингвистика и лингвистические базы данных: Докл. науч. конф. / Под ред. А. С. Герда. — СПб.: СПбГУ, 2002. С. 6–18.
15. *Vossen P.* Introduction to EuroWordNet // Computers Humanities, 1998. Vol. 32. No. 2–3. P. 73–89.
16. *Азарова И. В., Митрофанова О. А., Синопальникова А. А.* Компьютерный тезаурус русского языка типа WordNet // Мат-лы конф. «Диалог-2003». — М., 2003.
17. *Кожунова О. С.* Технология разработки семантического словаря системы информационного мониторинга: Автореф. дисс. . . . канд. техн. наук. — М.: ИПИ РАН, 2009. 2 с.
18. *Зацман И. М., Дурново А. А.* Моделирование процессов формирования экспертных знаний для мониторинга программно-целевой деятельности // Информатика и её применения, 2011. Т. 5. Вып. 4. С. 84–98.
19. *Kozhunova O.* Lexical and semantic methods in design of the problem-oriented linguistic resources // WORLDCOMP'11: 2011 World Congress in Computer Science, Computer Engineering and Applied Computing Proceedings. — Las Vegas: CSREA Press, 2011. Vol. II. P. 618–624.
20. *Kozhunova O.* Cross-disciplinary approach to expert activity cognitive interoperability support // 5th Conference (International) on Cognitive Science Proceedings. — Kaliningrad, 2012. С. 91–92.
21. *Кожунова О. С.* Моделирование лексической семантики в задачах компьютерной лингвистики // Системы и средства информатики, 2012. Т. 22. № 1. С. 86–109.
22. *Kozhunova O.* Detection of nominalized structures in parallel patent texts in Russian and in German // WORLDCOMP'09: 2009 World Congress in Computer Science, Computer Engineering and Applied Computing Proceedings. — Las Vegas: CSREA Press, 2009. Vol. I. P. 479–485.
23. *Кожунова О. С.* Выявление номинализованных конструкций в параллельных текстах патентных документов на русском и немецком языках // Компью-

терная лингвистика и интеллектуальные технологии: По мат-лам ежегодной Междунар. конф. «Диалог'2009». — М.: РГГУ, 2009. Вып. 8(15). С. 185–191.

24. *Дулин С. К., Дулина Н. Г., Кожунова О. С.* Когнитивная интероперабельность экспертной деятельности и ее приложение в геоинформатике // Конф. по искусственному интеллекту (КИИ-2012): Труды 13-й Национальной конф. по искусственному интеллекту с международным участием. — Белгород: БГТУ им. В. Г. Шухова, 2012. С. 351–357.
25. *Дулин С. К., Розенберг И. Н.* О развитии методологических основ и концепций геоинформатики // Системы и средства информатики. Спец. вып.: Научно-методологические проблемы информатики. — М.: ИПИ РАН, 2006. С. 201–256.
26. *Цветков В. Я.* Информатизация, инновационные процессы и геоинформационные технологии // Геодезия и аэрофотосъемка, 2006. № 4. С. 112–118.
27. *Кошелев А. Д.* Концептуально-смысловая модель образования лексической полисемии // 5-я Междунар. конф. по когнитивной науке: Тезисы докладов. — Калининград, 2012. Т. 2. С. 464–465.
28. *Norvig P., Lakoff G.* Taking: A study in lexical network theory // 13th Berkeley Linguistics Society Annual Meeting Proceedings: BLS, 1987. P. 195–206.
29. *Виноградов В. В.* Основные типы лексических значений слова // Избранные труды. Лексикология и лексикография. — М., 1977. С. 162–189.
30. *Buddenberg R.* Toward an interoperability reference model, 2006. http://web1.nps.navy.mil/?budden/lecture.notes/interop RM.html.
31. *Черниговская Т. В., Дубасова А. В., Риехакайнен Е. И.* Лексическая неоднозначность и организация ментального лексикона // 5-я Междунар. конф. по когнитивной науке: Тезисы докладов. — Калининград, 2012. Т. 2. С. 698–700.
32. *Кузнецов О. П.* О возможности организации знаний на основе когнитивной семантики // 5-я Междунар. конф. по когнитивной науке: Тезисы докладов. — Калининград, 2012. Т. 2. С. 806–807.
33. *Lakoff J.* Women, fire, and dangerous things: What categories reveal about the mind. — University of Chicago Press, 1987.
34. *Зайцев Д.* Язык как зеркало мышления: Рецензия на книгу Джорджа Лакоффа «Женщины, огонь и опасные вещи: что категории языка говорят нам о мышлении» / Пер. с англ. И. Б. Шатуновского. — М.: Языки славянской культуры, 2004. 792 с. // Отечественные записки, 2004.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 94–101

# РАЗРАБОТКА ИМИТАЦИОННОЙ МОДЕЛИ СБОРА И ОБРАБОТКИ ДАННЫХ ЭКСПЕРИМЕНТОВ НА УСКОРИТЕЛЬНОМ КОМПЛЕКСЕ НИКА*

В. В. Кореньков[1], А. В. Нечаевский[2], В. В. Трофимов[3]

**Аннотация:** В работе обоснована необходимость создания имитационной модели системы хранения и обработки данных ускорительного комплекса НИКА. В качестве платформы для создания модели выбрана система GridSim. В работе описан подход к моделированию системы хранения данных dCache и каналов передачи. На простом примере показаны возможности использования модели.

**Ключевые слова:** грид-технологии; грид-инфраструктуры; система хранения данных; оптимизация; моделирование; исследование; разработки; dCache; Tier1; НИКА; грид

## 1 Введение

В настоящее время в Объединенном институте ядерных исследований создается ускорительный комплекс НИКА.

Комплекс НИКА представляет собой ускоритель тяжелых ионов НИКА и установку MPD (multipurpose detector), объединяющую детекторы для изучения ядерной материи в горячем и плотном состоянии, которое возникает при столкновении ускоренных тяжелых ионов. Установка MPD является источником данных с интенсивностью потока десятки петабайт в год.

Ожидаемая интенсивность потока данных настолько велика, что массивы данных характеризуются как сверхбольшие. Для обработки таких потоков данных используются распределенные системы коллективного пользования, построенные на грид-технологиях.

Для оптимизации структуры будущего комплекса обработки данных необходимо определить его основные параметры, структуру и проверить предлагаемые технические решения с помощью моделирования. Для этих целей на базе пакета моделирования GridSim создана имитационная модель грид-сайта.

## 2 Система обработки данных ускорительного комплекса НИКА

Хранение и использование экспериментальных данных в современных исследованиях в области физики высоких энергий является актуальной проблемой. Объем получаемых и обрабатываемых данных исключает возможность их хранения и использования не только на одном кластере, но и в пределах одной организации, поэтому на первый план выходит создание распределенной системы хранения и обработки данных.

Для эксперимента MPD на НИКА предполагается, что поток данных будет иметь следующие параметры:

- высокая скорость набора событий (до 6 кГц);
- в центральном столкновении Au-Au при энергиях НИКА образуется до 1000 заряженных частиц;
- размер файла с первоначальной моделируемой информацией с детекторов для одного события занимает около 0,45 МБ.

*Работа частично выполнена в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007–2013 годы» (гос. контракт № 07.524.12.4008).

[1]Объединенный институт ядерных исследований, korenkov@cv.jinr.ru

[2]Объединенный институт ядерных исследований, Andrey.Nechaevskiy@gmail.com

[3]Объединенный институт ядерных исследований, trofimov@jinr.ru

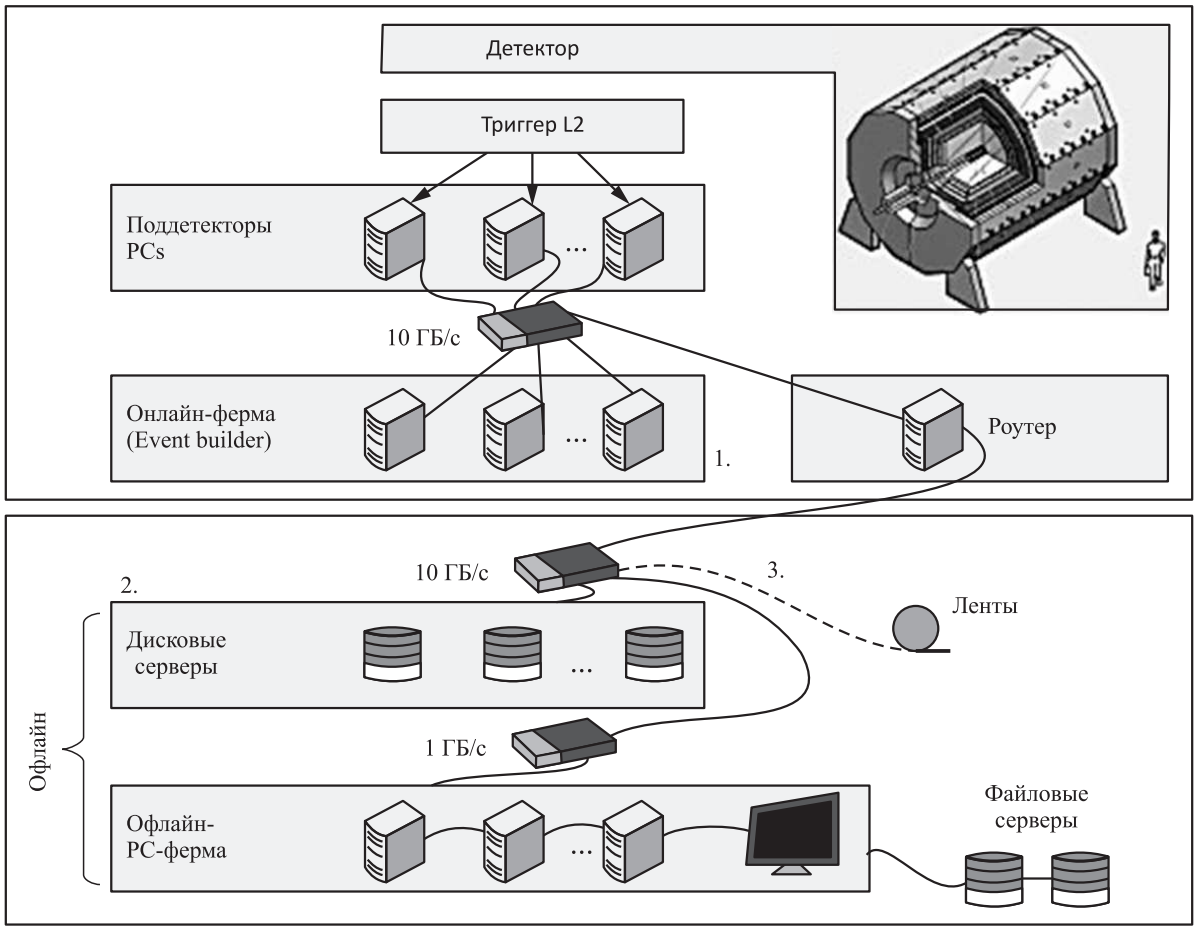



**Рис. 1** Схема обработки физических данных ускорительного комплекса НИКА

Схема получения и обработки данных представлена на рис. 1.

Данные, идущие от персональных компьютеров поддетекторов MPD, накапливаются специально предназначенными для сборки событий программами (Event Builder) компьютерной фермы в режиме онлайн. После формирования события в режиме офлайн через специально предназначенную для этой цели волоконно-оптическую линию связи с пропускной способностью 10 ГБ/с данные записываются на диск.

После триггера высокого уровня отобранные события записываются в RAW-файлы (скорость записи один файл в 1 минуту сбора данных) и затем полностью восстанавливаются.

Прогнозируемое число обрабатываемых событий при этом составляет приблизительно $19 \cdot 10^9$. Принимая скорость передачи данных от датчиков равной 4,7 ГБ/с, общий объем исходных данных можно оценить в 30 ПБ ежегодно, или 8,4 ПБ после сжатия. Эти оценки основаны на особенностях DAQ (data acquisition) и подобных оценках, выполненных для эксперимента ALICE [1].

В качестве системы обработки физической информации в эксперименте НИКА предполагается использование грид. Грид (название по аналогии с электрическими сетями — electric power grid) — это компьютерная инфраструктура нового типа, обеспечивающая глобальную интеграцию информационных и вычислительных ресурсов. Суть инициативы грид состоит в создании набора стандартизированных служб для обеспечения надежного, совместимого, дешевого и безопасного доступа к географически распределенным высокотехнологичным информационным и вычислительным ресурсам — отдельным компьютерам, кластерам и суперкомпьютерным центрам, хранилищам информации, сетям, научному инструментарию и т. д. [2].

Эксперименты, в которых для обработки данных используется грид-инфраструктура или облачные вычисления, имеют некоторые общие черты:

**Таблица 1** Уровни иерархической модели и их функции [3]

| Уровень | Функции |
|---|---|
| Tier0 | Первичная реконструкция событий, калибровка, хранение копий полных баз данных |
| Tier1 | Полная реконструкция событий, хранение актуальных баз данных по событиям, создание и хранение наборов анализируемых событий, моделирование, анализ |
| Tier2 | Репликация и хранение наборов анализируемых событий, моделирование, анализ |

**Таблица 2** Функции и свойства симуляторов грид [7]

| Функция | GridSim | OptorSim | Monarc | ChicSim | SimGrid | MicroGrid |
|---|---|---|---|---|---|---|
| Репликация данных | Да | Да | Да | Да | Нет | Нет |
| Издержки записи/чтения диска | Да | Нет | Да | Нет | Нет | Да |
| Комплексное фильтрование или запросы данных | Да | Нет | Нет | Нет | Нет | Нет |
| Планировка пользовательских задач | Да | Нет | Да | Да | Да | Да |
| Резервирование центрального процессорного устройства | Да | Нет | Нет | Нет | Нет | Нет |
| Симуляция нагрузки | Да | Нет | Нет | Да | Нет | Нет |
| Дифференцированное качество обслуживания сети | Да | Нет | Нет | Нет | Нет | Нет |
| Генерация фонового сетевого трафика | Да | Да | Нет | Нет | Да | Да |

большие потоки данных, длительный цикл проектирования и строительства, длительный период эксплуатации. Так, компьютерная инфраструктура для эксперимента ALICE представляет собой иерархическую грид-структуру с компьютерными центрами класса Tier 0/1/2. Функциональные различия уровней иерархической модели представлены в табл. 1. Для хранения и обработки данных в эксперименте PANDA [4] также предполагается использование грид.

Проектирование грид-структур больших масштабов подразумевает не только привлечение специалистов, обладающих уникальными навыками, но и применение инструментов для моделирования. При создании распределенной системы требуется принять решения по архитектуре инфраструктуры, количеству ресурсных центров, объему требуемых ресурсов. Кроме того, необходимо обеспечить достаточную пропускную способность, решить проблемы сохранности данных, обеспечить распределение ресурсов между различными группами пользователей, выбрать алгоритмы обработки и запуска задач и многое другое. Для решения этих вопросов, а также обоснования решений требуется создание имитационной модели обработки данных эксперимента. Возникает необходимость создания имитационной модели, которая бы удовлетворяла всем условиям.

Актуальность темы обусловливается тем, что на основе модели в дальнейшем могут быть обоснованы рекомендации и техническое задание на разработку компьютерной инфраструктуры, рассмотрены различные варианты организации хранения данных эксперимента.

## 3 Выбор пакета моделирования

На сегодняшний день существуют различные инструменты моделирования грид-систем [5]. Проект GridSim разрабатывается группой исследователей в лаборатории по изучению облачных и распределенных вычислений отдела информатики и компьютерных вычислений в Университете Мельбурна, Австралия. Пакет моделирования GridSim неоднократно применялся [6] для моделирования грид-структур и планировщиков.

GridSim — это библиотека классов, предназначенных для построения модели грид-системы. Она, в свою очередь, построена на стандартной библиотеке SimJava, с помощью которой можно моделировать поток дискретных событий во времени. Приложение создается расширением классов GridSim и объединением их в программу, которая моделирует обработку потока заданий грид-структурой, обладающей определенными ресурсами и с заданной дисциплиной их резервирования и использования. В сравнении с другими пакетами моделирования грид GridSim обладает рядом преимуществ. Основные преимущества представлены в табл. 2.

С помощью GridSim можно проводить воспроизводимые эксперименты, которые сложно реализовать в настоящем окружении динамических грид-систем.

После анализа целого ряда систем для разработки имитационной модели была выбрана платформа GridSim.

# 4 Моделирование сайта уровня T1 грид-структуры

В качестве примера грид-структуры уровня T1 будет рассмотрен Офлайн-уровень обработки физических данных ускорительного комплекса НИКА. Для эффективной работы грид-сайта, проведения исследований по оптимизации нагрузки, разработки и тестирования новых алгоритмов с точки зрения скорости достижения результата необходимо использовать средства моделирования грид-систем. При создании модели предполагается, что основой для построения системы хранения данных будет dCache [8]. Модель сайта T1 строится на следующем алгоритме обработки данных (см. рис. 1):

(1) данные появляются с заданной частотой и записываются на локальные диски компьютеров. После перемещения данных на второй уровень диск очищается;

(2) данные перемещаются автоматически на второй уровень по каналам. В качестве носителей второго уровня используются пулы системы dCache, рассматриваемые в модели как единая память. При обработке данных предполагается, что вначале данные попадают в дисковый пул системы хранения, а затем по локальному протоколу передается на узлы обработки. Непосредственное монтирование директории на рабочих узлах не используется;

(3) для долгосрочного хранения данных используется ленточный робот. Копии файлов автоматически создаются на лентах, после чего файлы удаляются с дисковых пулов.

Отличительная особенность конфигурации dCache — наличие не менее двух уровней хранения: жесткие диски и ленточный накопитель. Под ленточным накопителем подразумеваются автоматизированные библиотеки, оснащенные роботизированным загрузочным механизмом и стойкой на несколько картриджей (лент). Объем такой библиотеки $Q$ можно определить простейшими вычислениями, исходными данными для которых будут производительность установки $p$, время ее работы $T$ и емкость накопителей $c$:

$$Q = \frac{pT}{c}\,.$$

Другие вопросы создания грид-сайта требуют более тщательного анализа и выбора приемлемого варианта. Таким образом, перед разработчиками системы встают следующие вопросы:

- определение необходимого количества драйвов;
- способы группировки файлов на лентах;
- политика записи файлов.

Стоит отметить ряд ключевых особенностей GridSim, которые потребовали доработки из-за несоответствия требованиям модели:

- создавать файлы может только пользователь;
- все объекты моделирования объединены в сеть при помощи каналов передачи данных;
- пользователь может копировать (создавать) только один файл единовременно.

Для решения этих вопросов потребовалось расширение существующих классов и добавление новых объектов. Так, в систему добавлены следующие объекты (рис. 2):

- Drive — драйв магнитофона;
- Arm — рука робота;
- Reel archive — архив картриджей;
- Reel — картридж.

Набор этих классов позволяет моделировать все процессы, происходящие с копией файла на лентах: загрузку и выгрузку ленты манипулятором, монтирование на драйве, поиск файла на ленте и его чтение/запись.

Задача моделирования сетевой инфраструктуры в библиотеке GridSim решена с помощью классов Router, Link, NetPacket и некоторых других. Этот набор средств позволяет моделировать прохождение пакетов по сети. Пользователю предоставляется возможность встраивать свои планировщики пакетов в исходную модель. Такой подход обеспечивает высокую точность моделирования. Его недостатком применительно к задаче моделирования T1 является избыточность — вопросы маршрутизации, столкновений пакетов, влияния фоновой загрузки каналов в данной модели не рассматриваются и, следовательно, уровень детализации до пакета представляется избыточным. В рассматриваемом случае интерес представляет только изменение нагрузки на отдельные компоненты сети.

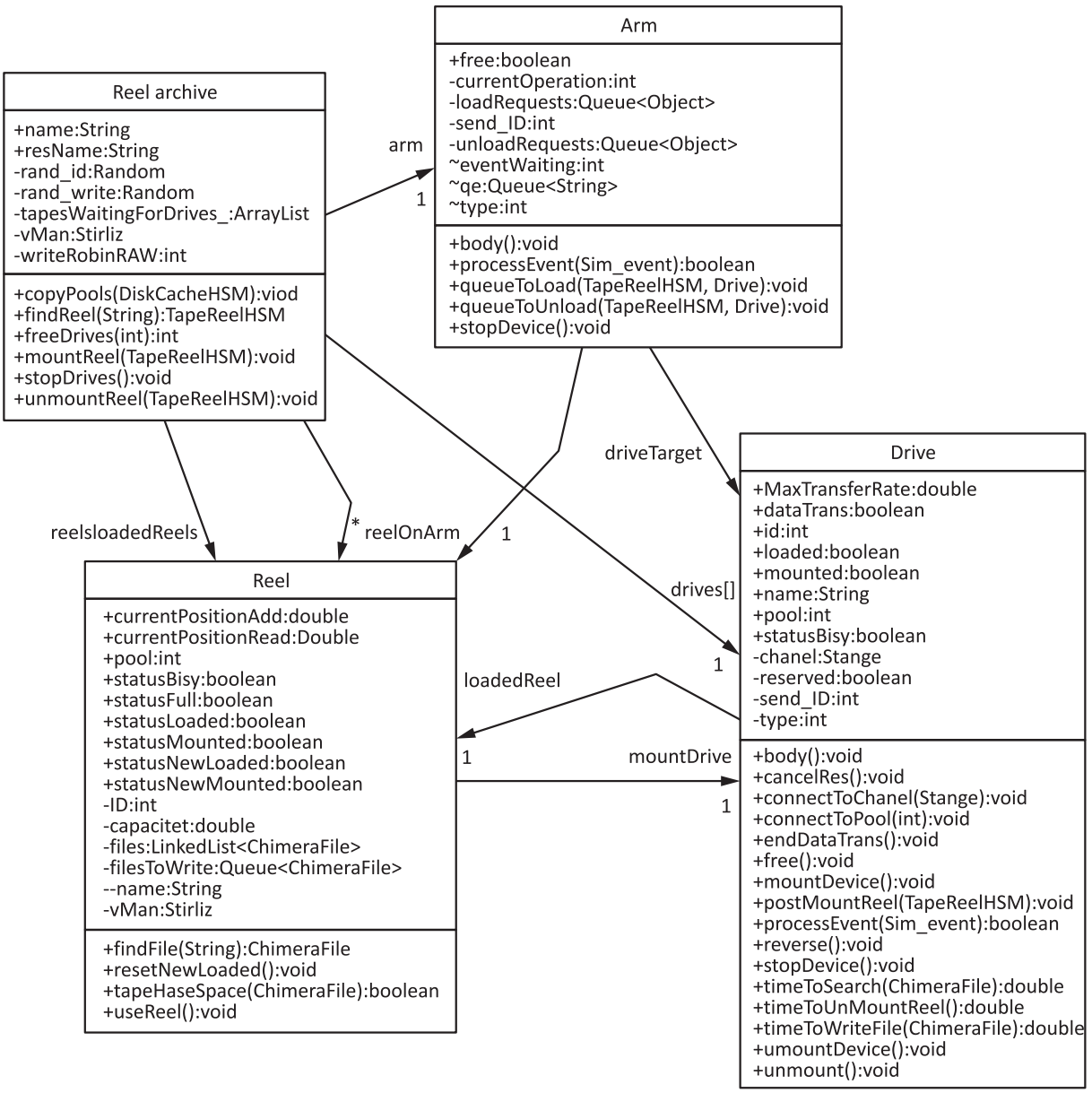



**Рис. 2** Описание новых классов в модели

Исходя из вышеизложенного требуется дополнить GridSim следующим механизмом.

Вводится понятие *операции передачи данных*. Под этим подразумевается запись/чтение части или целого файла экспериментальных данных. В этом случае ввод и вывод служебной и диагностической информации считается пренебрежимо малым. Операция рассматривается как атомарная, т. е. начинается методом «начать операцию». Параметрами метода являются: устройство 1 — источник данных, устройство 2 — получатель и список всех промежуточных устройств, которые необходимо пройти от источника до получателя. Элемент сети в системе описывается классом Stange. Взаимодействие классов, описывающих сеть, отражено на рис. 3.

Результаты моделирования доступны пользователю в виде таблиц и графиков. Для этой цели используются классы генератора лога и визуального отображения результатов: Info — описание вычислительной структуры и потока заданий; Reporter — генератор лога; парсер лога; объект визуального отображения результатов и др.

Ниже приведем пример задачи, которая возникает при проектировании системы сбора и хранения данных.

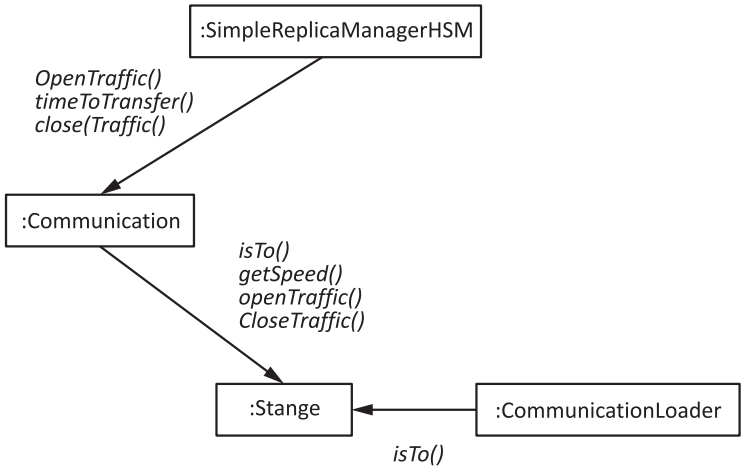



**Рис. 3** Взаимодействие классов, описывающих сеть

## 5 Пример использования модели

С помощью системы моделирования можно исследовать прохождение набора заданий и передачу файлов через грид-структуру с заданной пользователем топологией и параметрами центров обработки. Модель позволяет получить оценку временны́х параметров обработки потока заданий при заданной пользователем дисциплине распределения ресурсов между заданиями и структурой очередей к центрам обработки.

Моделирование дает ответы на вопросы:

- какие вычислительные ресурсы требуются для обработки данных;
- как должны быть связаны между собой центры обработки;
- каким должен быть уровень сжатия данных;
- какой должна быть конфигурация роботизированной библиотеки;
- хватит ли ресурсов на обработку потока данных и предоставление данных пользователям.

Проиллюстрировать применение упомянутых выше классов можно на примере моделирования процесса обработки данных с одновременной записью на ленты. Задача проектировщика — определить необходимое количество драйвов библиотеки. При этом исследуются два вопроса: какое количество драйвов библиотеки необходимо для того, чтобы записать весь поток «сырых» (RAW) данных с детекторов эксперимента, и насколько при этом процесс обработки данных (поток заданий от пользователей) будет мешать записи, если обработка потребует загрузки файлов с лент на диски.

Допустим, что имеется в распоряжении библиотека, количество драйвов в библиотеке фиксировано и равно пяти. Это существенно меньше необходимого, но достаточно для иллюстрации возможностей модели. Когда для обслуживания поступающих на сайт заданий и записи RAW-данных используются одни и те же пулы (драйвы), процесс начинает вести себя хаотично, многократно монтируя и размонтируя ленты для записи даже при незначительных загрузках. Для того чтобы избежать этой ситуации, в рассматриваемой модели пулы лент разделены на принимающие данные (RAW) и обслуживающие поток заданий (DLT — digital linear tape). Возникает вопрос: каким образом распределить драйвы между двумя пулами при фиксированных параметрах потока заданий? Предполагаем, что файлы запрашиваются случайным образом.

Моделируемая система — двухуровневая. На первом уровне находится дисковый массив, на втором уровне — ленточный накопитель. В существующей модели скорость записи и чтения с дискового массива не зависит от загрузки. Параметры драйвов и робота соответствуют параметрам планируемых к установке устройств (табл. 3). Количество драйвов в роботе фиксировано, и есть только одна «рука», загружающая файлы в драйв.

Результаты моделирования приведены в табл. 4. С помощью модели исследовались следующие характеристики:

- время выполнения — астрономическое время выполнения потока заданий, которое из общих соображений будет уменьшаться с увеличением количества драйвов;
- длина очереди — максимальная длина очереди на запись RAW-данных на ленту.

**Таблица 3** Параметры для моделирования ленточной библиотеки

| Параметр | Значение |
|---|---|
| Время монтирования/размонтирования, с | 22 |
| Скорость поиска, с | 300 |
| Скорость чтения/записи, с | 120 |
| Скорость перемотки, с | 1000 |
| Время загрузки/разгрузки картриджа в драйв, с | 100 |
| Размер файла, МБ | 6000 |

**Таблица 4** Результаты моделирования

| Эксперимент | Драйвов RAW | Драйвов DLT | Время выполнения, с | Длина очереди |
|---|---|---|---|---|
| 1 | 1 | 4 | 28 959 | 13 |
| 2 | 2 | 3 | 28 703 | 1 |
| 3 | 3 | 2 | 28 814 | 1 |
| 4 | 4 | 1 | 59 275 | 1 |

Моделирование показало, что при заданном темпе сбора данных для записи должно быть выделено не менее двух драйвов. С другой стороны, для обработки потока заданий должно быть выделено не менее двух драйвов для чтения накопленной информации. Если за критерий оптимальности принять минимальное астрономическое время выполнения потока заданий, то оптимальным можно считать распределение драйвов по варианту № 2.

Этот пример иллюстрирует один из вариантов использования программы. Такие исследования могут быть проведены с использованием аналитических моделей теории массового обслуживания, однако добавление простейших условий группировки заданий и файлов значительно усложняет аналитические модели, тогда как для имитационной модели изменения сводятся к нескольким строчками программного кода.

# 6 Заключение

Созданная система моделирования позволяет проводить разнообразные эксперименты с исследуемым объектом, не прибегая к физической реализации. Это позволяет предсказать и предотвратить большое число неожиданных ситуаций в процессе эксплуатации, которые могли бы привести к неоправданным затратам, потере данных, а возможно, и к повреждению дорогостоящего оборудования. В процессе моделирования можно подобрать минимально необходимое оборудование, обеспечивающее потребности передачи, обработки и хранения данных, оценить необходимый запас производительности оборудования, обеспечивающего возможное увеличение производственных потребностей, выбрать несколько вариантов оборудования с учетом текущих потребностей и перспективы развития в будущем, провести проверку работы системы, выявить ее «узкие» места и т. д.

Применение системы моделирования позволит определить параметры системы обработки данных ускорительного комплекса НИКА на этапе технического проектирования.

Дальнейшее развитие системы предполагает внесение дополнений с целью создания модели грид-сайта уровня T1 с использованием двух и трех уровней dCache. Для моделирования предполагается использовать оригинальный алгоритм назначения пулов dCache и оригинальные данные по потокам. Также необходимо провести полномасштабные испытания модели с целью выявления ошибок и создания базы сценариев моделирования.

Важное значение разработанной системы моделирования связано с созданием в Объединенном институте ядерных исследований автоматизированной системы обработки и хранения данных (АСОД) уровня T1 для эксперимента CMS (Compact Muon Solenoid) на Большом адронном коллайдере и предназначенной для работы в составе глобальной грид-системы для обработки данных (WLCG — Worldwide LHC Computing Grid). Автоматитзированная система обработки и хранения данных нацелена на проведение полного цикла обработки физической информации, получаемой в ходе проведения эксперимента, обеспечения работ по моделированию физических процессов, защищенного хранения и приема/передачи данных в другие центры WLCG. Основной системой хранения данных в АСОД является dCache. Очевидно, что в процессе длительного (10 лет и более) функционирования центра будет необходимо оперативно масштабировать систему хранения и повышать эф-

фективность использования ленточного робота в системе dCache без остановки работы всего комплекса. В этом процессе предварительное моделирование работы системы хранения станет необходимым инструментом.

Результаты работы могут быть рекомендованы для использования при проектировании грид-системы для сбора, передачи, обработки и хранения данных с мегаустановок или других аналогичных установок, генерирующих большие объемы данных.

## Литература


1. *Cortese P., Carminati F., Fabjan C. W., et al.* ALICE Technical Design Report of the Computing // CERN/LHCC 2005-018, ALICE TDR 12, 2005.
2. *Кореньков В. В.* Грид-технологии: статус и перспективы // Вестник Международной академии наук. Русская секция, 2010. № 1. С. 41–44.
3. *Ильин В. А., Кореньков В. В., Солдатов А. А.* Российский сегмент глобальной инфраструктуры LCG // Открытые системы, 2003. № 1. С. 56–60.
4. Веб-портал проекта PANDA. http://www-panda.gsi.de.
5. *Нечаевский А. В., Кореньков В. В.* Пакеты моделирования DataGrid // Системный анализ в науке и образовании: Электронный журнал, 2009. № 1.
6. Веб-портал проекта GridSim. http://www.gridbus.org/gridsim.
7. *Sulistio A., Cibej U., Venugopal S., Robic B., Buyya R.* A toolkit for modelling and simulating data grids: An extension to GridSim // Concurrency Computation Practice Experience (CCPE), 2008. Vol. 20. No. 13. P. 1591–1609.
8. Веб-портал проекта dCache. http://www.dcache.org.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 102–105

# ОЦЕНКИ СКОРОСТИ СХОДИМОСТИ РАСПРЕДЕЛЕНИЙ НЕКОТОРЫХ СЛУЧАЙНЫХ СУММ К УСТОЙЧИВЫМ ЗАКОНАМ*

В. Ю. Королев[1], Л. М. Закс[2]

**Аннотация:** Приведены оценки скорости сходимости распределений специальных сумм случайного числа независимых одинаково распределенных случайных величин с конечными дисперсиями к симметричным строго устойчивым законам. Предполагается, что случайный индекс имеет смешанное пуассоновское распределение, в котором смешивающее распределение является устойчивым законом, сосредоточенным на положительной полуоси. Абсолютные константы выписаны в явном виде.

**Ключевые слова:** устойчивое распределение; неравенство Берри–Эссеена; случайная сумма; дважды стохастический пуассоновский процесс (процесс Кокса); смешанное пуассоновское распределение

Функцию распределения и плотность строго устойчивого распределения с характеристическим показателем $\alpha$ и параметром $\theta$, задаваемого характеристической функцией

$$\mathfrak{g}_{\alpha,\theta}(t) = \\ = \exp\left\{-|t|^{\alpha}\exp\left\{-\frac{i\pi\theta\alpha}{2}\mathrm{sign}t\right\}\right\}, \ t\in\mathbb{R}, \quad (1)$$

где $0<\alpha\leq 2$, $|\theta|\leq\theta_{\alpha}=\min\{1, 2/\alpha-1\}$, будем обозначать соответственно $G_{\alpha,\theta}(x)$ и $g_{\alpha,\theta}(x)$. Симметричным строго устойчивым распределениям соответствует значение $\theta=0$. Односторонним устойчивым распределениям соответствуют значения $\theta=1$ и $0<\alpha\leq 1$. Функцию распределения и плотность стандартного нормального закона ($\alpha=2$, $\theta=0$) будем обозначать соответственно $\Phi(x)$ и $\phi(x)$:

$$\phi(x)=\frac{1}{\sqrt{2\pi}}\,e^{-x^2/2}; \quad \Phi(x)=\int\limits_{-\infty}^{x}\phi(z)\,dz\,.$$

Рассмотрим последовательность независимых одинаково распределенных случайных величин $X_1, X_2, \ldots$, заданных на некотором вероятностном пространстве $(\Omega, \mathfrak{A}, \mathsf{P})$. Будем предполагать, что

$$\mathsf{E}X_1=0\,, \ \ 0<\sigma^2=\mathsf{D}X_1<\infty\,.$$

Для натурального $n\geq 1$ положим

$$S_n=X_1+\cdots+X_n\,.$$

Пусть $N_1, N_2, \ldots$ — последовательность целочисленных неотрицательных случайных величин, заданных на том же самом вероятностном пространстве так, что при каждом $n\geq 1$ случайная величина $N_n$ независима от последовательности $X_1, X_2, \ldots$ Всюду далее для определенности будем считать, что $\sum\limits_{j=1}^{0}=0$.

Принято считать, что случайная последовательность $N_1, N_2, \ldots$ неограниченно возрастает ($N_n\longrightarrow\longrightarrow\infty$) по вероятности, если для любого $m\in(0,\infty)$ $\mathsf{P}(N_n\leq m)\longrightarrow 0$ при $n\to\infty$. Всюду далее символы $\Longrightarrow$ и $\stackrel{d}{=}$ обозначают соответственно сходимость по распределению и совпадение распределений.

В статье [1] доказан следующий критерий сходимости сумм случайного числа независимых одинаково распределенных случайных величин *с конечными дисперсиями* к симметричным строго устойчивым законам.

**Лемма 1.** *Предположим, что случайные величины $X_1, X_2, \ldots$ и $N_1, N_2, \ldots$ удовлетворяют указанным выше условиям, причем $N_n\longrightarrow\infty$ по вероятности при $n\to\infty$. Для того чтобы при $n\to\infty$*

$$\mathsf{P}\left(\frac{S_{N_n}}{\sigma\sqrt{n}}<x\right)\Longrightarrow G_{\alpha,0}(x)\,,$$

*необходимо и достаточно, чтобы*

$$\mathsf{P}(N_n<nx)\Longrightarrow G_{\alpha/2,1}(x)\,.$$

*Работа поддержана Российским фондом фундаментальных исследований (проекты 12-07-00115а, 12-07-00109а, 11-01-00515а и 11-07-00112а).

[1]Факультет вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; Институт проблем информатики РАН, vkorolev@cs.msu.su

[2]Альфа-банк, отдел моделирования и математической статистики, lily.zaks@gmail.com

В лемме 1 главным условием является сходимость распределений нормированных индексов $N_n$ к одностороннему строго устойчивому распределению $G_{\alpha/2,1}(x)$. Далее будет рассматриваться довольно полезная с точки зрения практических приложений специальная ситуация, в которой это условие выполнено.

В книге [2] предложено моделировать эволюцию неоднородных хаотических стохастических процессов, в частности динамику цен финансовых активов, с помощью обобщенных дважды стохастических пуассоновских процессов (обобщенных процессов Кокса). Этот подход получил дополнительное обоснование и развитие в книгах [3–6]. В книгах [6, 7] этот подход успешно применен к моделированию процессов плазменной турбулентности. В соответствии с указанным подходом поток информативных событий, в результате каждого из которых появляется очередное «наблюденное» значение рассматриваемой характеристики, описывается с помощью точечного случайного процесса вида $M(\Lambda(t))$, где $M(t)$, $t \geq 0$, — однородный пуассоновский процесс с единичной интенсивностью, а $\Lambda(t)$, $t \geq 0$, — независимый от $M(t)$ случайный процесс, обладающий следующими свойствами: $\Lambda(0) = 0$, $\mathsf{P}(\Lambda(t) < \infty) = 1$ для любого $t > 0$, траектории $\Lambda(t)$ не убывают и непрерывны справа. Процесс $M(\Lambda(t))$, $t \geq 0$, называется дважды стохастическим пуассоновским процессом (процессом Кокса). В частности, если процесс $\Lambda(t)$ допускает представление

$$\Lambda(t) = \int_0^t \lambda(\tau)\, d\tau\,, \quad t \geq 0\,,$$

в котором $\lambda(t)$ — положительный случайный процесс с интегрируемыми траекториями, то $\lambda(t)$ можно интерпретировать как мгновенную стохастическую интенсивность процесса Кокса.

В соответствии с такой моделью в каждый момент времени $t$ распределение случайной величины $M(\Lambda(t))$ является смешанным пуассоновским. Для большей наглядности рассмотрим случай, когда в рассматриваемой модели время $t$ остается фиксированным, а $\Lambda(t) = nU_{\alpha/2,1}$, где $n$ — вспомогательный параметр, $U_{\alpha/2,1}$ — случайная величина с функцией распределения $G_{\alpha/2,1}(x)$, независимая от стандартного пуассоновского процесса $M(t)$, $t \geq 0$. При этом асимптотика $n \to \infty$ может интерпретироваться как то, что (случайная) интенсивность потока информативных событий считается очень большой. Для каждого натурального $n$ положим

$$N_n = M(nU_{\alpha/2,1})\,.$$

Очевидно, что так определенная случайная величина $N_n$ имеет смешанное пуассоновское распределение:

$$\mathsf{P}(N_n = k) = \mathsf{P}\left(M(nU_{\alpha/2,1}) = k\right) = \\ = \int_0^\infty e^{-nz}\frac{(nz)^k}{k!}\, g_{\alpha/2,1}(z)\, dz\,, \quad k = 0, 1, \ldots \quad (2)$$

Случайная величина $N_n$ может быть интерпретирована как число событий, зарегистрированных к моменту времени $n$ в пуассоновском процессе со случайной интенсивностью, имеющей строго устойчивую плотность $g_{\alpha/2,1}(z)$. Высокая адекватность устойчивых распределений как моделей статистических закономерностей динамики цен финансовых активов отмечается во многих работах (см., например, [8]).

Предположим, что случайная величина $U_{\alpha/2,1}$ и пуассоновский процесс $M(t)$ независимы от последовательности $X_1, X_2, \ldots$ Тогда, очевидно, при каждом $n$ случайная величина $N_n$ также будет независима от этой последовательности.

Обозначим $A_n(z) = \mathsf{P}(N_n < nz)$, $z \geq 0$ ($A_n(z) = 0$ при $z < 0$). Несложно видеть, что

$$A_n(x) \Longrightarrow G_{\alpha/2,1}(x) \ \ (n \to \infty)\,.$$

Действительно, как известно, если $\Pi(x;\ell)$ — функция распределения Пуассона с параметром $\ell > 0$ и $E(x;c)$ — функция распределения с единственным единичным скачком в точке $c \in \mathbb{R}$, то

$$\Pi(\ell x;\ell) \Longrightarrow E(x;1) \ \ (\ell \to \infty)\,.$$

Так как для $x \in \mathbb{R}$

$$A_n(x) = \int_0^\infty \Pi(nx;nz)\, dG_{\alpha/2,1}(z)\,,$$

то по теореме Лебега о мажорируемой сходимости при $n \to \infty$

$$A_n(x) \Longrightarrow \int_0^\infty E(x/z;1)\, dG_{\alpha/2,1}(z) = \\ = \int_0^x dG_{\alpha/2,1}(z) = G_{\alpha/2,1}(x)\,,$$

т. е. так определенные случайные величины $N_n$ удовлетворяют условию, фигурирующему в лемме 1.

В дополнение к сформулированным выше условиям на случайные величины $X_1, X_2, \ldots$ предположим, что

$$\beta^3 = \mathsf{E}|X_1|^3 < \infty\,. \quad (3)$$

Обозначим

$$D_{n,\alpha} = \sup_x \left| \mathsf{P}\left(S_{N_n} < x\sigma\sqrt{n}\right) - G_{\alpha,0}(x) \right| .$$

**Теорема 1.** *Пусть выполнены условия* (2) и (3). *Для любого* $n \geq 1$ *справедлива оценка*

$$D_{n,\alpha} \leq 0{,}2428 \frac{\Gamma\left(1/\alpha\right)\beta^3}{\alpha\sigma^3\sqrt{n}} .$$

Доказательство. Распределение случайной величины $N_n$ является смешанным пуассоновским (см. (2)). Следовательно, по теореме Фубини

$$\mathsf{P}\left(S_{N_n} < x\sigma\sqrt{n}\right) = \mathsf{P}\left(S_{M(nU_{\alpha/2,1})} < x\sigma\sqrt{n}\right) = \\ = \int\limits_0^\infty \mathsf{P}\left(S_{M(nz)} < x\sigma\sqrt{n}\right) g_{\alpha/2,1}(z)\, dz . \quad (4)$$

Далее, как известно, симметричное строго устойчивое распределение с параметром $\alpha$ является масштабной смесью нормальных законов, в которой смешивающим распределением является односторонний устойчивый закон ($\theta = 1$) с параметром $\alpha/2$:

$$G_{\alpha,0}(x) = \int\limits_0^\infty \Phi\left(\frac{x}{\sqrt{z}}\right) dG_{\alpha/2,1}(z) , \quad x \in \mathbb{R} \quad (5)$$

(см., например, теорему 3.3.1 в [9]). Из (4) и (5) следует, что

$$D_{n,\alpha} \leq \\ \leq \int\limits_0^\infty \sup_x \left| \mathsf{P}\left(\frac{S_{M(nz)}}{\sigma\sqrt{n}} < x\right) - \Phi\left(\frac{x}{\sqrt{z}}\right) \right| dG_{\alpha/2,1}(z) = \\ = \int\limits_0^\infty \sup_x \left| \mathsf{P}\left(\frac{S_{M(nz)}}{\sigma\sqrt{nz}} < x\right) - \Phi(x) \right| dG_{\alpha/2,1}(z). \quad (6)$$

Подынтегральное выражение в (6) оценим с помощью следующего аналога неравенства Берри–Эссеена для пуассоновских случайных сумм.

**Лемма 2.** *Пусть случайные величины* $X_1, X_2, \ldots$ *одинаково распределены, причем* $\mathsf{E}X_1 = 0$ *и* $\mathsf{E}|X_1|^3 < \infty$. *Пусть* $N_\lambda$ *— пуассоновская случайная величина с параметром* $\lambda > 0$. *Предположим, что случайные величины* $N_\lambda, X_1, X_2, \ldots$ *независимы в совокупности. Обозначим*

$$Z_\lambda = X_1 + \cdots + X_{N_\lambda} .$$

*Тогда*

$$\sup_x \left| \mathsf{P}\left(\frac{Z_\lambda}{\sqrt{\mathsf{D}Z_\lambda}} < x\right) - \Phi(x) \right| \leq \frac{0{,}3041}{\sqrt{\lambda}} \frac{\mathsf{E}|X_1|^3}{(\mathsf{E}X_1^2)^{3/2}} .$$

Доказательство этого утверждения приведено в [10], также см. теорему 2.4.3 в [5].

Далее понадобится следующее утверждение, позволяющее вычислить $\mathsf{E}U_{\alpha/2,1}^{-1/2}$, несмотря на то что плотность $g_{\alpha/2,1}(z)$, вообще говоря, нельзя выписать в явном виде в терминах элементарных функций.

**Лемма 3.**

$$\mathsf{E}U_{\alpha/2,1}^{-1/2} = \frac{\sqrt{2}\Gamma\left(1/\alpha\right)}{\alpha\sqrt{\pi}} .$$

Доказательство. Из (1) вытекает, что характеристическая функция симметричного ($\theta = 0$) строго устойчивого распределения имеет вид:

$$\mathfrak{f}_{\alpha,0}(t) = e^{-|t|^\alpha} , \quad t \in \mathbb{R} . \quad (7)$$

С другой стороны, записав соотношение (5) в терминах характеристических функций с учетом (7), получим

$$e^{-|t|^\alpha} = \int\limits_0^\infty \exp\left\{-\frac{t^2 z}{2}\right\} g_{\alpha/2,1}(z)\, dz . \quad (8)$$

Обозначим

$$h_{\alpha/2}(z) = \frac{\alpha}{\Gamma(1/\alpha)}\sqrt{\frac{\pi}{2}}\,\frac{g_{\alpha/2,1}(z)}{\sqrt{z}} , \quad z \geq 0 .$$

Обобщенным распределением Лапласа принято называть абсолютно непрерывное распределение вероятностей, задаваемое плотностью

$$\ell_\alpha(x) = \frac{\alpha}{2\Gamma(1/\alpha)}\, e^{-|x|^\alpha} , \quad -\infty < x < \infty .$$

Переобозначив аргумент $t \mapsto x$ и выполнив несколько формальных тождественных преобразований равенства (8), будем иметь:

$$\ell_\alpha(x) = \frac{\alpha}{2\Gamma(1/\alpha)} e^{-|x|^\alpha} = \\ = \frac{\alpha}{\Gamma(1/\alpha)}\sqrt{\frac{\pi}{2}} \int\limits_0^\infty \frac{\sqrt{z}}{\sqrt{2\pi}} \exp\left\{-\frac{x^2 z}{2}\right\} \frac{g_{\alpha/2,1}(z)}{\sqrt{z}}\, dz = \\ = \int\limits_0^\infty \sqrt{z}\phi(x\sqrt{z}) h_{\alpha/2}(z)\, dz . \quad (9)$$

Можно убедиться, что $h_{\alpha/2}(z)$ — плотность распределения неотрицательной случайной величины. Действительно, при каждом $z > 0$

$$\int\limits_{-\infty}^{\infty} \sqrt{z}\phi(x\sqrt{z})\,dx = 1\,.$$

Поэтому из (9) вытекает, что

$$1 = \int\limits_{-\infty}^{\infty} \ell_\alpha(x)\,dx = \int\limits_{-\infty}^{\infty}\int\limits_{0}^{\infty} \sqrt{z}\phi(x\sqrt{z})h_{\alpha/2}(z)\,dzdx = \\ = \int\limits_{0}^{\infty} h_{\alpha/2}(z)\left(\int\limits_{-\infty}^{\infty} \sqrt{z}\phi(x\sqrt{z})\,dx\right)\,dz = \\ = \int\limits_{0}^{\infty} h_{\alpha/2}(z)\,dz.$$

Лемма доказана.

Продолжив (6) с учетом лемм 2 и 3, получим

$$D_{n,\alpha} \le 0{,}3041\,\frac{\beta^3}{\sigma^3\sqrt{n}}\,\mathsf{E}U_{\alpha/2,1}^{-1/2} = \\ = 0{,}3041\,\frac{\sqrt{2}\Gamma\left(1/\alpha\right)}{\alpha\sqrt{\pi}}\,\frac{\beta^3}{\sigma^3\sqrt{n}}\,.$$

Теорема доказана.

# Литература


1. *Королев В. Ю.* О сходимости распределений случайных сумм независимых случайных величин к устойчивым законам // Теория вероятностей и ее применения, 1997. Т. 42. Вып. 4. С. 818–820.
2. *Gnedenko B. V., Korolev V. Yu.* Random summation: Limit theorems and applications. — Boca Raton: CRC Press, 1996.
3. *Bening V., Korolev V.* Generalized Poisson models and their applications in insurance and finance. — Utrecht: VSP, 2002. 434 p.
4. *Королев В. Ю., Соколов И. А.* Математические модели неоднородных потоков экстремальных событий. — М.: ТОРУС ПРЕСС, 2008.
5. *Королев В. Ю., Бенинг В. Е., Шоргин С. Я.* Математические основы теории риска. — 2-е изд., перераб. и доп. — М.: Физматлит, 2011. 620 с.
6. *Королев В. Ю.* Вероятностно-статистические методы декомпозиции волатильности хаотических процессов. — М.: Изд-во Московского ун-та, 2011. 510 с.
7. Stochastic models of structural plasma turbulence / Eds. V. Korolev, N. Skvortsova. — Utrecht: VSP, 2006. 400 p.
8. *McCulloch J. H.* Financial applications of stable distributions // Handbook of statistics. — Amsterdam: Elsevier Science, 1996. Vol. 14. P. 393–425.
9. *Золотарев В. М.* Одномерные устойчивые распределения. — М.: Наука, 1983.
10. *Korolev V., Shevtsova I.* An improvement of the Berry–Esseen inequality with applications to Poisson and mixed Poisson random sums // Scandinavian Actuarial J., 2012. Vol. 2012. No. 2. P. 81–105. Available online since June 4, 2010.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 106–113

# УНИВЕРСАЛЬНЫЙ МЕТРИЧЕСКИЙ ТЕЗАУРУС РУССКОГО ЯЗЫКА

Л. А. Кузнецов[1], В. Ф. Кузнецова[1], А. В. Капнин[2]

**Аннотация:** Известные тезаурусы русского языка составлены группами экспертов. В статье предлагается вариант разработки инструментов для автоматизированного формирования тезауруса на основе формального представления текстов, поясняющих семантику слов, и количественной оценки семантического расстояния между словами как меры их близости. Предлагаемые решения позволяют ориентироваться на формально-математические представления, минимизирующие элемент субъективности в оценке близости слов. Они открывают возможность синтеза автоматических систем оценки семантической близости слов и решения иных задач в области обработки текстов.

**Ключевые слова:** компьютерная лингвистика; универсальный тезаурус; метрический тезаурус; семантическая оценка близости; семантическое расстояние; теория информации

## 1 Введение и постановка задачи

Исследование возможностей создания систем автоматической оценки степени семантического подобия текстов, представленных на естественном (русском) языке, выдвинуло на первый план задачу оценки семантической близости двух любых слов, в том числе и разных частей речи. Решение задачи может базироваться на современных представлениях автоматизированной обработки текстовой информации, обеспечиваемых базами данных, способными быстро обрабатывать и хранить большие объемы слов. В общем случае такие базы данных называются тезаурусами. Тезаурус (от греч. сокровище) в современной лингвистике — особая разновидность словарей общей или специальной лексики, в которых указаны семантические отношения (синонимы, антонимы, паронимы, гипонимы, гиперонимы и т. п.) между лексическими единицами [1].

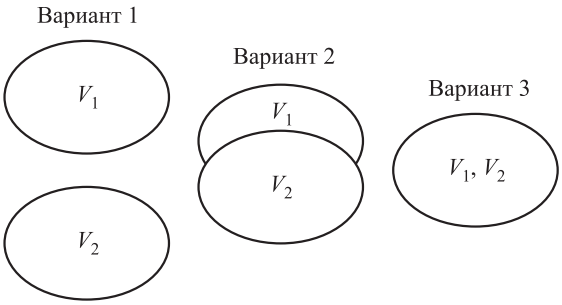
Известные тезаурусы русского языка (РуТез, RussNet, Лингвокультурный тезаурус русского языка и др.) составлены группами экспертов на основе их представлений о степени семантической близости отдельных слов. Такие тезаурусы вследствие неавтоматизированной методики их формирования, опирающейся на субъективные представления экспертов, как правило, имеют небольшой объем (количество статей и отношений между словами), не могут быть адаптированы к изменениям словарной базы. В них не могут быть введены отношения между словами различных частей речи, изменения словарной базы влекут их полную переработку. Связи в существующих тезаурусах выражены с помощью логических отношений, что мешает применению математического аппарата при создании и совершенствовании инструментов компьютерной лингвистики [2–4]. Мнения экспертов субъективны, поэтому даже однотипные семантические отношения (синонимы, антонимы, паронимы, гипонимы и т. п.) определяются различным образом представителями разных школ. Этот результат согласуется с теорией информации, в соответствии с которой вербальное представление информации всегда, с одной стороны, избыточно, а с другой — неоднозначно.

В математике используются метрические пространства, в которых определена метрика: расстояние между элементами этого пространства — точками. Простой пример — обычная комната (трехмерное пространство), в которой можно измерить расстояние между любыми точками, например электрической лампочкой, свисающей с потолка, и стаканом, стоящим на столе.

Представляется полезным и для оценки близости слов разработать некоторую формальную меру «расстояния» между словами, в соответствии с которой между словами, близкими по смыслу, расстояние было бы небольшим, а между словами, далекими по смыслу, — большим, т. е. чтобы расстояние отражало степень семантического подобия слов. При этом важно, чтобы мера имела объективную формально-математическую основу,

[1]Российская академия народного хозяйства и государственной службы при Президенте РФ (Липецкий филиал), kuznetsov.leonid48@gmail.com

[2]Липецкий государственный технический университет, gert@inbox.ru




Возможные варианты близости семантического содержания слов

а следовательно, допускала разработку автоматизированных процедур ее вычисления на основе имеющихся словарей русского языка без привлечения субъективных представлений экспертов.

Наглядная иллюстрация развиваемой далее идеи может быть получена на основании представлений теории множеств. Допустим, исследуется близость слов $V_1$ и $V_2$, содержание каждого из которых разъясняется в соответствующих словарных статьях словаря. Рисунок показывает возможное соотношение между содержанием словарных статей $V_1$ и $V_2$, которое интуитивно используется экспертами при оценке близости сопоставляемых слов. Вариант 1 соответствует случаю слов, имеющих совершенно различное семантическое содержание, при втором варианте имеет место частичная эквивалентность семантического содержания слов, а в третьем варианте слова $V_1$ и $V_2$ семантически эквивалентны.

Семантика слов полностью определяется соответствующими словарными статьями, на основании содержания которых эксперты приходят к выбору одного варианта из приведенных на рисунке. Содержательная сущность задачи оценки семантической близости слов показывает, что для ее автоматизации необходимо формализовать, во-первых, представление словарного описания содержания сопоставляемых слов $V_1$ и $V_2$, т. е. представить его в виде некоторого математического объекта, допускающего количественную оценку. Во-вторых, необходимо ввести некоторую формальную меру для оценки близости семантического содержания слов на основании определяющих их словарных статей, представленных в виде математических объектов. Положительное решение этих вопросов позволит разработать автоматическую процедуру оценки «семантического расстояния» между словами, опираясь исключительно на сведения существующих словарных описаний слов $V_1$ и $V_2$.

В настоящей статье предлагается вариант формально-математического представления текстов, поясняющих семантику слов, и методики оценки семантического расстояния между словами как меры их близости. Отличительной особенностью предлагаемых решений является ориентация на формальные представления, открывающие возможность синтеза на их основе автоматических систем оценки семантической близости слов и решения иных задач в области обработки текстов.

## 2 Формирование образа слова

Исследования показали, что для решения задачи оценки семантической близости слов может быть привлечен математический аппарат теории информации, который позволяет синтезировать характеристику, интуитивно представляющуюся адекватной мере «семантического расстояния» между словами. В работе [5] показано, что для оценки семантической близости вербально представленных объектов могут быть использованы элементы теории информации. Дальнейшие исследования позволили выявить возможность формирования на ее основе количественной меры «семантического расстояния» между словами.

Теория информации позволяет синтезировать математическую модель описания содержания слов для оценки меры их семантической близости. Получаемые на ее основе представления, с одной стороны, полностью согласуются с интуитивными представлениями о близости смысла слов, а с другой — позволяют ввести меру в виде количества информации. Семантика слов может быть извлечена исключительно из словарей соответствующего языка. В дальнейшем для определенности имеется в виду русский язык, хотя разрабатываемая мера близости применима и к другим языкам, которые в рассматриваемом смысле представляются более простыми из-за имеющихся в них дополнительных структурных ограничений.

Авторы разработали и реализовали оригинальную технологию определения численной меры семантической близости между всеми словами, на основе которой разработан **универсальный специфицированный** метрический тезаурус русского языка. Тезаурус наполняется автоматически по разработанной технологии. Объем тезауруса ограничен лишь объемом имеющихся в наличии словарей и может пополняться автоматически при появлении новых словарных ресурсов.

Полученный тезаурус решает не только поставленную задачу нахождения численной оценки смысловой близости слов, но и представляет наиболее полную базу отношений между словами

русского языка, которые позволяют автоматизировать кластеризацию слов (построение словарей) на основании логических отношений типа: синонимы и антонимы, гиперонимы и гипонимы и др., используя формальные методы, исключающие субъективизм экспертов.

Словари дают, по возможности, лаконичное, с минимальным количеством необязательных для отражения смыслового содержания слов разъяснение смысла. Составляются они специалистами, хорошо знающими язык и опирающимися на имеющийся набор предшествующих изданий. Поэтому использование в них замысловатых, экзотических или редких слов маловероятно.

В основу концепции положены две гипотезы:

(1) словарные статьи, определяющие смысл семантически близких слов, содержат больше одинаковых слов (в нормальной форме), чем семантически удаленных;

(2) чем больше словарных статей используется для определения каждого из слов, тем точнее можно оценить количественно степень их близости.

Первая гипотеза базируется на том, что сопоставляемые слова относятся к определенным частям речи. Различные части речи отражают вполне определенные семантические представления: объекты, состояния объектов, свойства объектов и т. п. Для пояснения близких семантических представлений используются однородные по морфологической и синтаксической структуре и словарному составу тексты. Поэтому логично ожидать, и это подтверждается дальнейшими исследованиями, что сходная семантика представляется близкими наборами слов. Чем дальше семантика слов, тем дальше структура и словарное представление текстов, поясняющих слова.

Вторая гипотеза обосновывается тем, что, например, в теории информации доказывается факт, что дополнительная информация об объекте не увеличивает его неопределенность. В контексте рассматриваемого вопроса это означает, что любая дополнительная информация о семантике слова не увеличивает ее неопределенность. Существуют различные словари русского языка: общие, предметно или профессионально ориентированные, специальные и пр., в каждом из которых можно найти несколько отличающиеся определения слов (словарных статей). Их объединение в базе данных не уменьшает информацию о слове, а, наоборот, увеличивает ее. При этом словарные статьи из использованных словарей могут быть структурированы и снабжены соответствующими пометами. В результате формируется *мультисловарь*, который сам по себе представляет культурную и научную ценность.

**Таблица 1** Значимые части речи

| Случайное событие | Тип |
|---|---|
| $A_1$ | Существительное |
| $A_2$ | Глагол |
| $A_3$ | Прилагательное |
| $A_4$ | Наречие |
| $A_5$ | Числительное |
| $A_6$ | Местоимение |
| $A_7$ | Предикатив (можно, пора) |
| $A_8$ | Причастие |
| $A_9$ | Деепричастие |

В соответствии со второй гипотезой формирование тезауруса начинается с объединения словарных статей из разных словарей для каждого слова в единое описание. Описание дополняется определяемым словом и трансформируется в набор слов. Набор слов фильтруется с помощью синтаксического анализатора [6]: остаются только значимые слова — это слова, часть речи которых свидетельствует о сильной смысловой нагрузке (табл. 1). Оставшиеся слова в наборе приводятся к нормальной форме (с помощью синтаксического анализатора [6]) и сортируются в алфавитном порядке. Далее из набора исключаются дубликаты, а полученные слова $w_1, w_2, \ldots$ сохраняются в базе данных через запятую в качестве образа $O = \{w_1, w_2, \ldots\}$ данного слова $V$.

Существует целое множество методов оценки семантических расстояний между словами [7]. Большинство из них применимо к сетевым структурам, когда между словами установлены логические отношения. Как было отмечено в начале статьи, допускающих компьютерную лингвистическую обработку тезаурусов не найдено. Иных методов формального представления описания семантического содержания слов, адаптированных для русского языка и допускающих введение количественных мер, обнаружено не было. Для адекватной оценки семантической близости слов на основании имеющихся их словарных описаний важно обеспечить возможность учета (отражения) морфологической и синтаксической структуры описаний слов. Структура описаний связана с грамматической принадлежностью описываемого слова; кроме того, различные слова, используемые в словарных статьях для разъяснения смысла определяемого слова, могут содержать различное количество информации и вносить различный вклад в меру близости слов. Структуризация позволяет учесть это при формировании количественной оценки меры близости. В работе [5] введено представление текста в виде математической модели случайного объекта:

$$M_\omega = \{\Omega, \aleph, P(A_j)\}, \qquad (1)$$

где $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$ — пространство элементарных исходов; $\aleph = \{A_1, A_2, \ldots, A_j, \varnothing, \Omega\}$ — алгебра событий $A_j$, составленных с помощью операций логического сложения, умножения и отрицания из элементарных событий $\omega_i$ и дополненная невозможным $\varnothing$ и достоверным $\Omega$ событиями; $P(A_j)$ — вероятности событий, составляющих алгебру, которые рассчитываются по вероятностям элементарных исходов $p(\omega_i)$, $i = 1, 2, \ldots, n_\omega$, составляющих это событие:

$$P(A_j) = \sum_{\omega \in A_j} p(\omega_i)\,. \qquad (2)$$

В данном контексте $\omega_i$, $i = 1, 2, \ldots, n$, — это слова, составляющие текст, а $n$ — их количество. Эмпирические вероятности $p(\omega_i)$, находящиеся под знаком суммы в (2), представляют относительные частоты, которые равны отношению числа появлений слова $i$ в тексте, к общему количеству слов в тексте, т. е. $p(\omega_i) = n_\omega/n$.

Алгебра событий $\aleph$ — это система, комбинируемая из используемых частей речи $A_j$. Введение алгебры определяет используемую структуризацию языка. Может быть использована морфологическая структуризация языка, в которой случайные события $A_j$, $j = 1, 2, \ldots, J$, отождествляются с частями речи, приведенными, например, в табл. 1.

Могут использоваться не все части речи, а, например, только знаменательные.

На основании вероятностей (2) событий $A_j$, $j = 1, 2, \ldots, J$, определяется энтропия текста, представленного в виде модели (1):

$$H_\omega = -\sum_{A_j \in \aleph} P(A_j) \ln P(A_j)\,. \qquad (3)$$

Модель текста, определяющего семантическое содержание конкретного слова, далее называется образом слова и обозначается $O$. После определения алгебры, по которой распределяются слова пояснительного текста, образ (1) конкретного слова содержит всю информацию о нем, включая и собственно количество информации, определяемое выражением (3).

## 3 Определение семантического расстояния между словами

Оценка близости двух слов $V_1$ и $V_2$, как отмечалось выше, может базироваться на введении некоторой меры расстояния между их образами $O_1$ и $O_2$. Введение меры расстояния может быть осуществлено в виде соотношения, характеризующего отношение общего количества информации, содержащейся в двух образах $O_1$ и $O_2$, к количеству взаимной информации [5], т. е. количеству информации, принадлежащей одновременно и $O_1$, и $O_2$. Такая мера может быть представлена в виде:

$$\rho_{1,2} = \frac{H_1^2 + H_2^1}{H^{1,2}}\,, \qquad (4)$$

где $H_{1,2}$, $H_1^2$, $H_2^1$ — энтропии, определяемые по образам сопоставляемых слов:

$$H_2^1 = -\sum_{A_j \in \aleph} P\left(A_{2j}^1\right) \ln P\left(A_{2j}^1\right)\,;$$

$$H_1^2 = -\sum_{A_j \in \aleph} P\left(A_{1j}^2\right) \ln P\left(A_{1j}^2\right)\,;$$

$$H^{1,2} = -\sum_{A_j \in \aleph} P\left(A_j^{1,2}\right) \ln P\left(A_j^{1,2}\right)\,.$$

Случайные события определяются следующим образом:

$$A_j^{1,2} = \{\omega_{ik} \in \Omega_1 + \Omega_2 | \omega_{ik} \in A_j^1 \cap \omega_{ik} \in A_j^2\}\,; \quad (5)$$

$$A_{2j}^1 = \{\omega_{ik} \in \Omega_1 + \Omega_2 | \omega_{ik} \in A_j^1 \cap \omega_{ik} \notin A_j^2\}\,; \quad (6)$$

$$A_{1j}^2 = \{\omega_{ik} \in \Omega_1 + \Omega_2 | \omega_{ik} \notin A_j^1 \cap \omega_{ik} \in A_j^2\}\,, \quad (7)$$

где верхним индексом отмечен объект, в который входит рассматриваемое слово, а нижним — объект, в который слово не входит.

Событие (5) составляется из слов, входящих в событие $A_j$ объектов 1 и 2; (6) объединяет слова, входящие в объект 1, но отсутствующие в объекте 2; (7) объединяет слова, не входящие в объект 1, но входящие в объект 2. Других вариантов для слов, присутствующих в двух сопоставляемых текстах, нет.

Так как энтропии вычисляются по вероятностям, то они являются безразмерными абстрактными величинами. Содержательный смысл им приписывается в соответствии с контекстом решаемых задач. В рассматриваемом случае семантическое расстояние может быть безразмерной величиной, принципиальным является диапазон изменения этого расстояния и непрерывность меры. Непрерывность меры (4) следует из непрерывности логарифмической функции. Диапазон изменения меры с учетом ее непрерывности может быть оценен по предельным (граничным) значениям расстояния, определяемого с ее помощью.

Допустим, что рассматриваются два объекта (слова из словаря). Обозначим их $V_1$ и $V_2$. Каждое из них сопровождается образом $O_i \leftrightarrow \{\omega_1, \omega_2, \ldots, \omega_n\}$, где $n$ — количество слов текста, определяющего смысл слова. Количество слов в определяющем тексте может быть различным: $n$, $m$, $l$ и т. п. Из содержательного представления вопроса о предельных значениях меры (4) понятно, что предельными являются такие варианты:

(1) образы слов полностью совпадают (вариант 3 на рисунке);

(2) образы слов полностью не совпадают (вариант 1 на рисунке).

В первом варианте $\{\omega_1, \omega_2, \ldots, \omega_n\}_1 = \{\omega_1, \omega_2, \ldots, \omega_n\}_2$; следовательно, будут совпадать распределения слов по событиям $A_j$, $j = 1, 2, \ldots, J$. В принципе не имеет значения, как распределятся слова по существительным, прилагательным, глаголам и т. д. Важно, что эти распределения будут полностью совпадать для первого и второго слова. События типа $A^1_{2j}$ и $A^2_{1j}$, $j = 1, 2, \ldots, J$, не будут содержать элементов (слов), поэтому вероятности $P(A^1_{2j}) = 0$ и $P(A^2_{1j}) = 0$, $j = 1, 2, \ldots, J$. Из этого следует, что и энтропии $H^1_2 = H^2_1 = 0$. Подставив эти значения в (4), можно получить величину расстояния между сопоставляемыми словами для случая полного совпадения поясняющих их текстов:

$$\rho_{1,2} = \frac{H^2_1 + H^1_2}{H^{1,2}} = \frac{0}{H^{1,2}} = 0\,.$$

По второму варианту $\{\omega_1, \omega_2, \ldots, \omega_n\}_1$ полностью отличается от $\{\omega_1, \omega_2, \ldots, \omega_n\}_2$, т. е. $\omega_{i1} \neq \omega_{i2}$ для всех $i = 1, 2, \ldots, n$. Следовательно, в этом случае не будет ни одного слова, которое встретилось бы одновременно в объяснении семантики и объекта $V_1$, и объекта $V_2$, поэтому случайные события системы (5) $A^{1,2}_j$, $j = 1, 2, \ldots, J$, не будут содержать реализаций и все вероятности $P(A^{1,2}_j)$, $j = 1, 2, \ldots, J$, будут равны нулю. В теории информации принято считать $0 \ln 0 = 0$. На основании этого и равенства нулю всех совместных вероятностей $P(A^{1,2}_j)$ следует равенство нулю энтропии $H^{1,2} = 0$. Тогда из (4) при произвольных значениях энтропий объектов 1, 2 следует:

$$\rho_{1,2} = \frac{H^2_1 + H^1_2}{H^{1,2}} = \frac{H^2_1 + H^1_2}{0} = \infty\,.$$

Таким образом, значение меры (4) семантического расстояния между словами изменяется от нуля при тождественном совпадении содержательного описания слов до бесконечности при полном отличии их описаний. Диапазон изменений и непрерывность меры (4) вполне соответствуют интуитивным представлениям о расстоянии вообще и семантическом расстоянии между словами в частности.

Расстояние (4) представляет мощный инструмент, который может быть использован в качестве количественной характеристики близости слов в различного рода системах, обрабатывающих текстовые источники.

# 4 Пример формирования образов слов и оценки семантического расстояния

Демонстрация практического применения методики осуществляется оценкой меры семантической близости между словом «Красивый» и словами «Прекрасный», «Великолепный», «Мечтательный». Слова выбраны так, что на основании общих представлений понятно, что семантическое расстояние между этими словами должно возрастать в порядке перечисления, а семантическое расстояние между одним и тем же словом, например «Красивый», должно быть нулевым.

В табл. 2 представлены структурированные по частям речи образы слов, являющиеся частью реальных образов. В примере алгебра состоит из трех событий $A_1$, $A_2$, $A_3$ (см. табл. 1), что является упрощением для наглядного представления технологии.

Покажем подробно процесс вычисления семантического расстояния между $V_1$ = «Красивый» и $V_2$ = «Прекрасный».

Объединенный образ для слов $V_1$ и $V_2$ выглядит так: {*Вид, Женщина, Линия, Очертание, Прелесть, Стан, Тон, Щеголь, Блестеть, Выглядеть, Казаться, Привлекать, Рисовать, Благовидный, Восхитительный, Красивый, Обворожительный, Прекрасный, Хороший,* ***Женщина, День, Прекрасница, Стан, Тон, Улыбка, Щеголь, Выглядеть, Казаться, Рисовать, Создавать, Украшать, Благовидный, Великолепный, Восхитительный, Замечательный, Красивый, Необыкновенный, Отличный, Прекрасный, Хороший***}. Количество слов в объединенном образе — 40.

События группы $A_1$ (существительные):

$A^1_{21}$ = {*Вид, Линия, Очертания, Прелесть*};

$A^2_{11}$ = {*День, Прекрасница, Улыбка*};

$A^{1,2}_1$ = {*Женщина, Стан, Тон, Щеголь, Женщина, Стан, Тон, Щеголь*}.

.

События группы $A_2$ (глаголы):

$A^1_{22}$ = {*Блестеть, Привлекать*};

$A^2_{12}$ = {*Создавать, Украшать*};

$A^{1,2}_2$ = {*Выглядеть, Казаться, Рисовать, Выглядеть, Казаться, Рисовать*}.

События группы $A_3$ (прилагательные):

$A^1_{23}$ = {*Обворожительный*};

$A^2_{13}$ = {*Великолепный, Замечательный; Необыкновенный, Отличный*};

**Таблица 2** Модели текстов, определяющих семантику сопоставляемых слов (образы слов)

| Слово | | $V_1$ = «Красивый» | $V_2$ = «Прекрасный» | $V_3$ = «Великолепный» | $V_4$ = «Мечтательный» |
|---|---|---|---|---|---|
| Алгебра | Событие $A_1$ (существительные) | Вид<br>Женщина<br>Линия<br>Очертание<br>Прелесть<br>Стан<br>Тон<br>Щеголь | ~~Женщина~~<br>День<br>Прекрасница<br>~~Стан~~<br>~~Тон~~<br>Улыбка<br>~~Щеголь~~ | Архитектура<br>~~Вид~~<br>Восклицание<br>Одобрение<br>~~Очертание~~<br>Радость | Взгляд<br>~~Вид~~<br>Мечтательность<br>Настроение |
| | Событие $A_2$ (глаголы) | Блестеть<br>Выглядеть<br>Казаться<br>Привлекать<br>Рисовать | ~~Выглядеть~~<br>~~Казаться~~<br>~~Рисовать~~<br>Создавать<br>Украшать | ~~Блестеть~~<br>Выражать<br>Выспаться<br>~~Рисовать~~ | Выражать<br>~~Смотреть~~<br>Создать<br>Стремиться |
| | Событие $A_3$ (прилагательные) | Благовидный<br>Восхитительный<br>Красивый<br>Обворожительный<br>Прекрасный<br>Хороший | ~~Благовидный~~<br>Великолепный<br>~~Восхитительный~~<br>Замечательный<br>~~Красивый~~<br>Необыкновенный<br>Отличный<br>~~Прекрасный~~<br>~~Хороший~~ | Величественный<br>Живописный<br>Картинный<br>~~Красивый~~<br>~~Прекрасный~~<br>Пышный<br>Роскошный | Возвышенный<br>Восхитительный<br>Мечтательный<br>Склонный |

**Примечание.** Зачеркнуты слова, которые присутствует в образе $V_1$ = «Красивый», с которым сравниваются другие слова.

$$A_3^{1,2} = \{\textit{Благовидный, Восхитительный, Красивый, Прекрасный, Хороший, Благовидный, Восхитительный, Красивый, Прекрасный, Хороший}\}\,.$$

Получаются энтропии:

$$H_2^1 = -\frac{4}{40}\ln\left(\frac{4}{40}\right) - \frac{2}{40}\ln\left(\frac{2}{40}\right) - \frac{1}{40}\ln\left(\frac{1}{40}\right) = 0{,}472\,;$$

$$H_1^2 = -\frac{3}{40}\ln\left(\frac{3}{40}\right) - \frac{2}{40}\ln\left(\frac{2}{40}\right) - \frac{4}{40}\ln\left(\frac{4}{40}\right) = 0{,}574\,;$$

$$H^{1,2} = -\frac{8}{40}\ln\left(\frac{8}{40}\right) - \frac{6}{40}\ln\left(\frac{6}{40}\right) - \frac{10}{40}\ln\left(\frac{10}{40}\right) = 0{,}953\,.$$

Семантическое расстояние между словами $V_1$ и $V_2$:

$$\rho_{1,2} = \frac{0{,}472 + 0{,}574}{0{,}953} = 1{,}098\,.$$

Результаты промежуточных расчетов и семантические расстояния представлены в табл. 3.

Для сравнения найдем семантические расстояние известным методом сопоставления свойств [7] (табл. 4), основанным на теоретико-множественном подходе Тверски, при котором слова образа понимаются как свойства:

$$\mu_{12} = \frac{|O_1| + |O_2|}{2|A^{1,2}|} - 1\,,$$

где $|O|$ — количество слов образа (мощность множества слов образа); $|A^{1,2}|$ — количество совпадающих слов в образах (мощность множества совпадающих слов образов).

Полученные в примере расстояния позволяют семантически позиционировать слова относительно друг друга. Предложенный метод, основанный на применении энтропии, показывает более содержательный результат, что закономерно объясняется использованием структуризации с помощью частей речи. Расстояние $\rho$ отражает объективную смысловую близость слов, а не просто выстраивает слова в порядке смысловой удаленности.

## 5 Краткая характеристика разработанного тезауруса

Отличительной особенностью разработанного тезауруса является универсальность, которая следует из того, что в нем ассимилированы сведения из 8 словарей четырех типов:

(1) словарь синонимов [8];

(2) словарь иностранных слов [9];

(3) толковые словари [10–14];

(4) энциклопедический словарь [15].

**Таблица 3** Семантические расстояния между словами, определенные с помощью энтропии

| Параметры | | Для $\rho_{11}$ с $V_1$ = = «Красивый» | Для $\rho_{12}$ с $V_2$ = = «Прекрасный» | Для $\rho_{13}$ с $V_3$ = = «Великолепный» | Для $\rho_{14}$ с $V_4$ = = «Мечтательный» |
|---|---|---|---|---|---|
| Общие | Расстояние $\rho_{1i}$ | 0 | 1,098 | 1,951 | 4,621 |
| | Всего слов | 38 | 40 | 36 | 31 |
| | Составляющая $H_2^1$ | 0 | 0,472 | 0,750 | 0,918 |
| | Составляющая $H_1^2$ | 0 | 0,574 | 0,679 | 0,716 |
| | Составляющая $H^{1,2}$ | 0,487 | 0,953 | 0,732 | 0,354 |
| Существительные | Количество событий $A_{i1}^1$ | 0 | 4 | 6 | 7 |
| | Количество событий $A_{11}^i$ | 0 | 3 | 4 | 3 |
| | Количество событий $A_1^{1,i}$ | 16 | 8 | 4 | 2 |
| Глаголы | Количество событий $A_{i2}^1$ | 0 | 2 | 3 | 4 |
| | Количество событий $A_{12}^i$ | 0 | 2 | 2 | 3 |
| | Количество событий $A_2^{1,i}$ | 10 | 6 | 4 | 2 |
| Прилагательные | Количество событий $A_{i3}^1$ | 0 | 1 | 4 | 6 |
| | Количество событий $A_{13}^i$ | 0 | 4 | 5 | 4 |
| | Количество событий $A_3^{1,i}$ | 12 | 10 | 4 | 0 |

**Таблица 4** Семантические расстояния между словами, определенные с помощью метода сопоставления свойств

| | Для $\mu_{11}$ с $V_1$ = «Красивый» | Для $\mu_{12}$ с $V_2$ = «Прекрасный» | Для $\mu_{13}$ с $V_3$ = «Великолепный» | Для $\mu_{14}$ с $V_4$ = «Мечтательный» |
|---|---|---|---|---|
| Количество слов образа, $\|O\|$ | 19 | 21 | 17 | 12 |
| Количество совпадающих слов в образах, $\|A^{1,2}\|$ | 19 | 12 | 6 | 2 |
| Семантическое расстояние, $\mu_{1i}$ | 0 | 0,375 | 0,917 | 2,5 |

Он содержит более 200 000 слов в нормальной форме и более 400 000 словарных статей, в которых отражаются различные семантические, морфологические и синтаксические характеристики слов.

Важнейшей сущностью тезауруса является интеллектуальная составляющая в виде отношений между словами, помещенными в тезаурус. Под отношением понимается метрическая оценка семантической близости слов. Интеллектуальная составляющая включает около 4 000 000 000 значимых отношений, численно выражающих семантическую близость слов.

При разработке актуальной версии тезауруса были использованы некоторые допущения, что неизбежно привело к соответствующим недостаткам:

– замена буквы «ё» на «е» в определяемых словах в ряде случаев приводит к трансформации в иное по смыслу слово, уже существующее в тезаурусе (например, нёбо → небо). Словарные статьи в этом случае объединяются, а данные слова становятся омонимами. Так как словарные статьи объединяются, то связи исходных «ё»-слов не будет утеряны, но численные оценки будут значительно занижены. Если учесть, что таких слов оказалось менее 0,1%, такое занижение оценок близости выглядит несущественным;

– при формировании образов тире заменяется на пробел, что вызывает и замену на пробел дефиса в силу того, что многие словари были оформлены ненадлежащим образом. С другой стороны, эта замена не должна привести к значимым колебаниям оценки, так как количество слов с дефисом не столь велико (менее 0,5%), а замена дефиса на пробел в большинстве случаев не меняет, а семантически расширяет образ слова;

– аббревиатуры и сокращения не расшифровывались; следовательно, синтаксический анализатор не мог корректно анализировать такие слова. Это приводило к семантическому искажению образов. Текст словарных статей преимущественно описывается без аббревиатур и

малым количеством сокращений слов (бóльшая часть которых не несет яркой смысловой нагрузки и будет исключена анализатором), что позволяет сделать вывод о небольшой значимости искажений;

– в силу ненадлежащего оформления текстов словарей возможны некоторые орфографические ошибки. Вручную проверить такой объем информации не представлялось возможным, поэтому текст подвергся автоматизированным проверкам орфографии.

Формирование тезауруса осуществлялось разработанной для этой цели автоматизированной системой. Хотя процесс и требует больших вычислительных ресурсов (высокопроизводительной вычислительной системы с клиент-серверной архитектурой и мощной дисковой подсистемой), но благодаря автоматизации можно обойти принципиальные препятствия на пути дальнейшего совершенствования тезауруса, направленного на исключение негативных последствий принятых допущений и недостатков, следующих из оформления текстов словарей. Спектр используемых в качестве основы тезауруса словарей может быть расширен и реализована более детальная синтаксическая и орфографическая проверка исходных текстов.

Несмотря на принятые допущения, уже разработанный тезаурус открывает широкое поле для работы в области компьютерной лингвистики. Численная оценка близости слов дает возможность применить математический аппарат в направлениях:

– кластеризации слов и формирования тезауруса на основе логических отношений;
– формирования словарей синонимов;
– построения мощных поисковых машин;
– сравнения и установления авторства текстов;
– разработки автоматических интерактивных систем консультирования и помощи, а также в других направлениях и областях обработки текстовой информации.

# 6 Заключение

Разработана методика представления словарных статей, разъясняющих содержание слов, в виде формальных объектов, имеющих количественную меру оценки содержащейся в них информации. Разработана мера оценки семантического расстояния между словами, позволяющая устанавливать степень их семантической близости. На их основе создана автоматизированная система формирования тезауруса русского языка на основании имеющихся словарей и разработан универсальный тезаурус, снабженный инструментом интеллектуальных отношений между словами, позволяющим оценивать их семантическую близость.

# Литература


1. *Лесников С. В.* Тезаурус как отражение системности языка // Вестник челябинского государственного ун-та, 2011. № 28 (243). Филология. Искусствоведение. Вып. 59. С. 52–61.
2. *Караулов Ю. Н.* Лингвистическое конструирование и тезаурус литературного языка. — М.: Наука, 1981. 367 с.
3. *Добров Б. В., Иванов В. В., Лукашевич Н. В., Соловьев В. Д.* Онтологии и тезаурусы: Учебно-методическое пособие. — Казань: КГУ, 2006. 190 с.
4. *Тарасов С. Д.* Подход к реализации автоматизированной системы построения тезауруса // Электронные библиотеки: перспективные методы и технологии, электронные коллекции: Труды IX Всеросс. научной конф. RCDL'2007. — Переславль-Залесский: Ун-т г. Переславля, 2007. Т. 2. С. 63–66.
5. *Кузнецов Л. А.* Вероятностно-статистическая оценка адекватности информационных объектов // Информатика и её применения, 2011. Т. 5. Вып. 4. С. 64–75.
6. *Антонова А. А., Мисюрев А. В.* Об использовании синтаксического анализатора Cognitive Dwarf 2.0 // Труды Института системного анализа РАН, 2008. № 38. С. 91–107.
7. *Крюков К. В., Панкова Л. А., Пронина В. А., Суховеров В. С., Шипилина Л. Б.* Меры семантической близости в онтологиях // Проблемы управления, 2010. № 5. С. 2–14.
8. *Абрамов Н.* Словарь русских синонимов и сходных по смыслу выражений. — М.: Русские словари, 1999. 431 с.
9. *Комлев Н. Г.* Словарь иностранных слов. — М.: ЭКСМО-Пресс, 2000. 1308 с.
10. *Винокур Г. О., Ларин Б. А., Ожегов С. И., Томашевский Б. В., Ушаков Д. Н.* Толковый словарь русского языка: в 4 т. — М.: Советская энциклопедия; ОГИЗ, 1935–1940. 5529 с.
11. Современный толковый словарь. — М.: Большая Советская энциклопедия, 1997. 6110 с.
12. *Даль В. И.* Толковый словарь живого великорусского языка. — М.: Цитадель, 1998. 4249 с.
13. *Ефремова Т. Ф.* Современный толковый словарь русского языка. — М.: АСТ, 2006. 3312 с.
14. *Ожегов С. И.* Толковый словарь русского языка. — М.: Оникс, 2008. 976 с.
15. Современный энциклопедический словарь. — М.: Большая Советская энциклопедия, 1997. 1382 с.

ИНФОРМАТИКА И ЕЁ ПРИМЕНЕНИЯ, 2013. Т. 7. Вып. 3. С. 114–125

# APPROXIMATION OF A MULTIDIMENSIONAL DEPENDENCY BASED ON A LINEAR EXPANSION IN A DICTIONARY OF PARAMETRIC FUNCTIONS*

M. G. Belyaev[1] and E. V. Burnaev[2]

**Abstract:** The problem of a multidimensional function approximation using a finite set of pairs "point" – "function value at this point" is considered. As a model for the function, an expansion in a dictionary containing nonlinear parametric functions has been used. Several subproblems should be solved when constructing an approximation based on such model: extraction of a validation sample, initialization of parameters of the functions from the dictionary, and tuning of these parameters. Efficient methods for solving these subproblems have been suggested. Efficiency of the proposed approach is demonstrated on some problems of engineering design.

**Keywords:** nonlinear approximation; parametric dictionaries

## 1 Introduction

For engineering design, it is necessary to model complex physical phenomena. Typically used models are represented by complex systems of differential equations. Such systems do not have analytical solutions; so, computationally heavy numerical methods are used. One approach to solving problems of engineering design, actively developing in recent years, is the surrogate modeling [1, 2]. In this approach, a complex physical phenomenon is described by a simplified (surrogate) model constructed using data mining techniques and a set of examples, representing results of a detailed physical modeling and/or real experiments. The problem of approximation of a multidimensional function using a finite set of pairs "point" – "value of the function at this point" is one of the main problems to be solved in the construction of the surrogate model. This problem will be considered in the following formulation:

**Problem 1.** *Let $f(\vec{x}) \in R^1$ be some continuous function on a compact set $\mathrm{D} \subset R^d$ with known output values in a finite set of input points. The set*

$$S_{\text{learning}} = \{\vec{x}_i, y_i\}_{i=1}^{N_{\text{learning}}}, \quad \vec{x}_i \in \mathrm{D}, \quad y_i = f(\vec{x}_i),$$

*forms the learning sample. The approximation problem is to construct an approximation $\hat{f}(\vec{x})$ (approximator) of the function $f(\vec{x})$ using the given data sample $S_{\text{learning}}$ such that $f(\vec{x}) \approx \hat{f}(\vec{x})$* for all $\vec{x} \in \mathrm{D}$.

**Remark 1.** In general case, the values of the function $f(\vec{x})$ are known only for the finite set of input points; so, the proximity $f(\vec{x}) \approx \hat{f}(\vec{x})$ is usually measured be the mean square error

$$Q\left(S_{\text{test}}, \hat{f}\right) = \left(\frac{1}{N_{\text{test}}}\right) \sum_{i=1}^{N_{\text{test}}} \left(y_i - \hat{f}(\vec{x}_i)\right)^2$$

calculated using an independent test data sample

$$S_{\text{test}} = \{\vec{x}_i, y_i\}_{i=1}^{N_{\text{test}}}, \quad \vec{x}_i \in \mathrm{D}, \quad y_i = f(\vec{x}_i).$$

The criterion $Q\left(S_{\text{test}}, \hat{f}\right)$ of approximation quality makes sense if input vectors from the samples $S_{\text{learning}}$ and $S_{\text{test}}$ are generated by the same distribution and cover the design space D sufficiently densely. The function $Q\left(S, \hat{f}\right)$ for some set $S$ of pairs "point" – "value of the function at this point" is called an error function on the set $S$.

Due to requirements of surrogate modeling problems (in particular, the need to build quickly computable global approximation model and to work with large data samples), the most common method of solving approximation problems is based on Artificial Neural Networks (ANN) [3]. An approximation based on the ANN model provides high-speed calculations of output predictions. The ANN model can be easily "extended" by increasing number of layers and/or their sizes as the learning sample size increases while the computational complexity of the approximation model construction grows only linearly.

A typical scheme of approximation construction based on the ANN model can be divided into two phases:

*The authors are partially supported by Laboratory for Structural Methods of Data Analysis in Predictive Modeling, MIPT, RF government grant, ag. 11.G34.31.0073; RFBR grant 13-01-00521. Results, described in this work, were obtained in the framework of "COPTI-X: Surrogate Model Construction for Structure Approximation and Optimization" joint project with Airbus Operations SAS.

[1]Institute for Information Transmission Problems RAS, Moscow Institute of Physics and Technology, Datadvance LLC, belyaev@iitp.ru

[2]Institute for Information Transmission Problems RAS, Moscow Institute of Physics and Technology, Datadvance LLC, burnaev@iitp.ru

(1) an initialization phase (setting the initial values of the model parameters; and

(2) extraction of the validation sample) and a training phase (tuning the ANN parameters to fit the data sample).

Usually, random methods are used during the initialization phase. Such methods lead to a large scatter of the approximation accuracy. Surrogate models are constructed to replace computationally heavy objective functions (and/or constraints) in optimization problems [4]. Engineering design based on surrogate models is iterative:

(1) find the optimum of the objective function (surrogate model);

(2) generate additional pairs "point" – "value of the function at the point" in the neighbourhood of the optimum;

(3) reconstruct the surrogate model and optimize it, etc.

Therefore, unpredictable significant changes in the surrogate model structure and significant variations of the approximation accuracy deteriorate surrogate based optimization.

This paper investigates methods for constructing approximations based on the linear expansion in nonlinear functions from the parametric dictionary (ANN model with one hidden layer). Here, the methods for initialization and training that reduce the average approximation error and its variations are suggested. Let describe the structure of the paper.

The model based on a linear expansion in a dictionary of parametric functions is described in subsection 2.1. The main steps of the approximation construction based on this model are described in subsection 2.2. Subsection 2.3 is devoted to various subproblems that arise when constructing an approximation using the proposed algorithm. In section 3, the subproblems of the initialization step of the approximation construction algorithm are considered. In subsection 3.1, a new algorithm is described for the validation sample extraction, such that points from the validation sample are distributed as uniformly as possible among the remaining points of the learning sample. A computationally efficient deterministic algorithm for the validation sample extraction based on the greedy optimization of some uniformity criterion. In subsection 3.3, a new algorithm is described for the initialization of functions from the parametric dictionary. In section 4, a new training algorithm is considered. In subsection 4.1, a special form of the error function that takes into account the structure of the approximation error dependence on different groups of parameters and includes adaptive regularization is suggested. In subsection 4.2, a new method for the regularization parameter selection is described. Both sections 3 and 4 contain results of the computational experiments on artificial functions. Experimental results for some engineering design problems are described in section 5.

# 2 Algorithm for Approximation Construction

## 2.1 Model based on a linear expansion in a dictionary of parametric functions

The approximation $\hat{f}(\vec{x})$ is modeled by the linear expansion in a dictionary of parametric functions, i. e.,

$$\hat{f}(\vec{x}) = \sum_{j=1}^{p} \alpha_j \psi_j\left(\vec{\theta}_j, \vec{x}\right) + \alpha_0 .$$

Let rewrite this equality in matrix form

$$\hat{f}(\vec{x}) = \vec{\psi}(\Theta, \vec{x})\,\vec{\alpha}$$

where the vectors are denoted by caps with vector signs and matrices are denoted by caps ($\vec{\alpha} = \{\alpha_j\}_{j=0}^{p}$, $\Theta = \{\vec{\theta}_j\}_{j=1}^{p}$). The row-vector $\vec{\psi}(\Theta, \vec{x})$ consists of the dictionary functions values at the point $\vec{x}$ ($\psi_0 \equiv 1$ corresponds to $\alpha_0$). Therefore, the approximator $\hat{f}(\vec{x})$ is defined by the matrix $\Theta$ of dictionary functions parameters and by the vector $\vec{\alpha}$ of the linear combination coefficients. In order to form the dictionary, sigmoid functions (sigmoids) are used:

$$\psi_j\left(\vec{\theta}_j, \vec{x}\right) = \sigma\left(\vec{x}^{\mathrm{T}}\theta_j + \theta_j^0\right);$$
$$\vec{\theta}_j = (\theta_j, \theta_j^0), \ \theta_j \in R^d, \ \theta_j^0 \in R^1,$$

where

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \ z \in R^1 .$$

The dictionary consisting of such functions can be used for approximation of rather wide class of functions $f$ (see the results in papers [5, 6]). For example, in [6], it is shown that the approximator $\hat{f}$, composed of $p$ sigmoid functions, allows to get the approximation accuracy of the order $O\left(1/p^{1/d}\right)$. However, if in the dictionary are not included arbitrary sigmoid functions but they are selected depending on the approximated function $f$ ("tune" the dictionary functions), then the approximation accuracy has the order $O(1/p)$.

## 2.2 Structure of the Approximation Algorithm

In order to construct the dictionary, it is necessary to select type and number of functions in the dictionary and initialize their parameters. It is impossible to determine the dictionary functions parameters explicitly. The standard approach approximation construction has been applied that uses partitioning of the original sample into two parts $S_{\text{learning}} = S_{\text{train}} \cup S_{\text{validation}}$ in order to prevent overtraining [7].

**Algorithm 1** (main steps of the approximation construction algorithm)

1. Model Selection Step (in the considered case, the model structure is defined by the number $p$ of the dictionary functions).
2. Initialization Step:
   (a) divide the sample $S_{\text{learning}}$ into subsamples $S_{\text{train}}$ and $S_{\text{validation}}$; and
   (b) set the initial values of the dictionary functions parameters $\Theta$ and expansion coefficients $\vec{\alpha}$.
3. Training Step:
   (a) iteratively minimize $Q\left(S_{\text{train}}, \hat{f}\right)$ with respect to the parameters $\Theta$ and $\vec{\alpha}$; and
   (b) stop minimization if the error $Q\left(S_{\text{validation}}, \hat{f}\right)$ begin to increase.

Each of the steps of Algorithm 1 is a separate subproblem. There exists a lot of methods for solution of these subproblems. In this paper, more efficient methods are proposed (except solution for the subproblem of the dictionary size $p$ selection). As for the choice of the dictionary size $p$, usually, some upper bound on its value is defined depending on the learning sample size and the exact value of $p$ is selected using cross-validation.

## 2.3 Optimization Algorithm

Usually, in order to optimize the error function (see step 3 of Algorithm 1) with respect to parameters of complex regression models (for example, multilayer neural networks), gradient methods are used due to very high dimensionality of the parametric space. Since in the considered case the number of parameters is relatively small, second-order optimization methods can be used. The Gauss–Newton method [8] is the most common method for minimizing functions of the form

$$Q\left(S_{\text{train}}, \hat{f}\right) = \left(\vec{y} - \hat{f}(X)\right)^{\mathrm{T}} \left(\vec{y} - \hat{f}(X)\right)$$

where $X$ is a matrix of all points from $S_{\text{train}}$ and $\vec{y} = f(X)$ is a vector of the function $f$ values at these points. In fact, the main difference between this method and the Newton method consists in how the matrix of second-order derivatives of the error function is calculated. Let denote by $\Omega = \{\Theta, \vec{\alpha}\}$ the set of all parameters of the model (parameters of the dictionary functions and the corresponding expansion coefficients); then, assuming that a residual vector components $\vec{e} = \hat{f}(X) - \vec{y}$ are small, one gets

$$Q_{\Omega\Omega} = \hat{f}_{\Omega}^{\mathrm{T}} \hat{f}_{\Omega} + \sum_{i=1}^{N_{\text{train}}} e_i \hat{f}_{\Omega\Omega}\left(\vec{x}_i\right) \approx \hat{f}_{\Omega}^{\mathrm{T}} \hat{f}_{\Omega} = J^{\mathrm{T}} J \quad (1)$$

where $J = \hat{f}_{\Omega}$ is the matrix of the model $\hat{f}$ derivatives with respect to $\Omega$ at the points $S_{\text{train}}$. Since $Q_{\Omega\Omega} \approx J^{\mathrm{T}} J \succeq 0$, then the approximate matrix of the second derivatives, calculated according to formula (1), is always nonnegative definite. This partially solves degeneracy problem of the Newton method being applied to nonconvex functions.

Nevertheless, let note that during the minimization of the error function approximation of the Hessian, $Q_{\Omega\Omega} \approx J^{\mathrm{T}} J$ can become degenerate and noninvertible.

The Levenberg–Marquardt algorithm [9] was developed to solve this degeneracy problem. The main idea is to add identity matrix with regularization multiplier to the approximate Hessian matrix when searching the step size according to the Gauss–Newton method, i. e.,

$$\Omega^k = \Omega^{k-1} - \left(J^{\mathrm{T}} J + \mu \mathrm{I}\right)^{-1} Q_{\Omega}\,. \quad (2)$$

The parameter $\mu$ defines behavior of the algorithm: for small $\mu$, the step size is close to the step size of the Gauss–Newton method; for big $\mu$, the step is done along the antigradient with the size approximately equal to $1/\mu$. Therefore, one can always find such value of the parameter $\mu$ which provides decrease of the error function. When training the model, the Levenberg–Marquardt algorithm will be used (this method is also realized in MatLab for ANN training).

# 3 Initialization Step

In this section, the algorithms for solving two subproblems of the initialization step are proposed: extraction of the validation sample and initialization of the model parameters. Also, in subsection 3.3, a methodology for an experimental comparison of approximation algorithms is described and the results of this comparison are provided.

## 3.1 Extraction of the Validation Sample

Let consider the first subproblem of the initialization step, i. e., decomposition of the initial sample $S_{\text{learning}}$ into two parts $S_{\text{train}}$ and $S_{\text{validation}}$ where $S_{\text{train}}$ is used for an iterative tuning of the model parameters and $S_{\text{validation}}$ is used to estimate a generalization ability (an estimate of the proximity between the original function and its approximation) and to stop the iterative tuning process when the overtraining appears.

In order to estimate the generalization, the set of points $X_{\text{validation}} = \{\vec{x} \in S_{\text{validation}}\}$ should be "uniformly" distributed among other points of the learning sample $X_{\text{learning}} = \{\vec{x} \in S_{\text{learning}}\}$. Standard algorithms for approximation construction perform decomposition into the validation $S_{\text{validation}}$ and the training $S_{\text{train}}$ samples randomly, which often results in inconsistent decomposition.

Let estimate the uniformity of $X_{\text{learning}}$ decomposition into the sets $X_{\text{train}}$ and $X_{\text{validation}}$ using the following criterion:

$$U(X_{\text{train}}, X_{\text{validation}}) = \frac{1}{\#r^1} \sum_{\{\vec{x}_i, \vec{x}_j\} \in r^1} \frac{1}{\|\vec{x}_i - \vec{x}_j\|} - \frac{1}{\#r^2} \sum_{\{\vec{x}_i, \vec{x}_j\} \in r^2} \frac{1}{\|\vec{x}_i - \vec{x}_j\|} \quad (3)$$

where $r^1$ is the set of pairs of points such that each of points belongs either to $X_{\text{train}}$ or to $X_{\text{validation}}$; $r^2$ is the set of pairs of points such that one of them belongs to $X_{\text{train}}$ and another belongs to $X_{\text{validation}}$; and $\#r^i$ is the cardinality of $r^i$, $i = 1, 2$. When minimizing $U(X_{\text{train}}, X_{\text{validation}})$ (with respect to different decompositions of $X_{\text{learning}}$), the distance between points from one class is maximized and the distance between points from different classes is minimized that fully meets all objectives.

Optimization of the criterion (3) is an $NP$-hard combinatorial problem that cannot be solved for reasonable time when the sample size $N_{\text{learning}} \gg 1$. On the other hand, for the case $N_{\text{learning}} \sim 10$, the optimization can be performed by the full search. Therefore, let consider the simplification of this optimization problem: divide all the design domain into rather small hypercubes and optimize the criterion locally by relocating points from one class ($S_{\text{train}}$) to another ($S_{\text{validation}}$) only within each of the hypercubes.

In order to divide the design domain, Classification and Regression Trees [10] have been used. This method is based on the sequence of simple cuts of the design domain with respect to the input vector components. In each of the hypercubes, obtained during the previous iteration, a constant approximation was constructed by averaging the output values of the points belonging to this hypercube. The input component and the location of the next cut are selected optimally in the sense of the mean square error of the corresponding piecewise constant approximation. There are a lot of criteria for stopping the tree construction process. These criteria are based on the generalization ability estimation of the tree [3]. Here, as a stopping criterion, an upper bound on the number of points belonging to each leaf of the tree is used since the optimization complexity of the criterion (3) depends on this upper bound.

**Algorithm 2** (sample $S_{\text{learning}}$ decomposition)

1. Construct a regression tree [10] with an upper bound on the number of points belonging to the tree leaves (the number of points should be bigger than $\text{leaf}_{\min} = 8$ and smaller than $\text{leaf}_{\max} = 16$) approximating the function $f$ with piecewise constant approximation. Let the number of leaves of the constructed tree be equal to some $K$.
2. Let $s_k$ be the set of points $\vec{x} \in X_{\text{learning}}$ belonging to the leaf with the number $k$.
3. For all $k = 1, \ldots, K$ using greedy algorithm (local optimization in each separate hypercube), decompose the set $s_k$ into subsets $s_k^{\text{val}}$ and $s_k^{\text{tr}}$ such that the criterion $U(s_k^{\text{val}}, s_k^{\text{tr}})$ takes its minimal value under the restriction that the fraction of the validation sample size is not smaller than $\text{val}_{\text{part}} = 0.2$.
4. Construct $S_{\text{validation}}$ and $S_{\text{train}}$

$$S_{\text{validation}} = \left\{ \{\vec{x}, y = f(\vec{x})\} : \vec{x} \in \bigcup_{k=1}^{K} s_k^{\text{val}} \right\};$$

$$S_{\text{train}} = \left\{ \{\vec{x}, y = f(\vec{x})\} : \vec{x} \in \bigcup_{k=1}^{K} s_k^{\text{tr}} \right\}.$$

The proposed algorithm contains two computationally expensive steps: construction of the regression tree (complexity is equal to $O(N_{\text{learning}} \log(N_{\text{learning}}))$) and local optimization of the criterion $U(s_k^{\text{val}}, s_k^{\text{tr}})$ (complexity is equal to $O(K) = O(N_{\text{learning}})$). For the second step, the constant in $O(K)$ depends on the number of ways to decompose a leaf with $\text{leaf}_{\text{num}} \in [\text{leaf}_{\min}, \text{leaf}_{\max}]$ points into two parts. Due to the restriction on the proportion between the sizes of the validation and the training samples, this number is equal to $C_{\text{leaf}_{\text{num}}}^{[\text{leaf}_{\text{num}} \cdot \text{val}_{\text{part}}]}$, i. e., it is bigger than 28 (for $\text{leaf}_{\text{num}} = \text{leaf}_{\min} = 8$) and smaller than 560 (for $\text{leaf}_{\text{num}} = \text{leaf}_{\min} = 16$).

## 3.2 Initialization of the Model Parameters

Initialization of the dictionary functions parameters $\Theta$ significantly influences the approximation construction process and the final approximation accuracy.

The random methods Nguyen–Widrow (NW) [11] and SCAWI (statistically controlled activation weight initialization) [12] are the most widely used algorithm for $\Theta$ initialization. These methods use some matrix of independent random variables, multiplied by a fixed factor defining smoothness of the dictionary functions.

Note that the random vectors generation in high-dimensional spaces leads to their clustering thus giving rise to clustering of the directions $\{\theta_j\}_{j=1}^p$. Here, an initialization algorithm, which generates a rich functional dictionary (in terms of a uniform distribution of the directions) is proposed. The directions $\{\theta_j\}_{j=1}^p$ are generated uniformly on the unit sphere using the methods from [13]. This method is based on a normalization of points generated by a multivariate normal distribution. The method uses the invariance property of a normal density relative to an arbitrary rotation and efficiently generates points with uniform distribution on the unit sphere.

Let now consider how to select norms values (defined by some scaling multipliers) of the vectors $\{\theta_j\}_{j=1}^p$. The value of the norm influences the smoothness of the corresponding sigmoid: for a sufficiently small value, the sigmoid is almost linear on the compact D; for a big value, the sigmoid behaves like the step function. If one fixed value is used for all norms, then all sigmoids will have equal smoothness and the dictionary will not be "rich" enough. Therefore, in order to define the norms values, the multipliers generated by the uniform distribution in some range have been used. Thus, the vectors $\{\theta_j\}_{j=1}^p$ have different norms and, finally, the dictionary contains sigmoids with different smoothness.

**Algorithm 3** (initialization of the parameters $\Theta$)

1. Construct a matrix $S$ of size $p \times d$ containing $p$ vectors, generated by the uniform distribution on a $d$-dimensional sphere of unit radius.
2. Let $r = \sqrt{d}\, p^{1/N_{\text{train}}}$ and generate values of scaling multipliers $\xi_j, j = 1, \ldots, p$, uniformly randomly on $[0, r]$. Such definition of $r$ guarantees that the number of points belonging to the domain of a sigmoid saturation is small [11].
3. Let define the sigmoids direction vectors $\{\theta_j\}_{j=1}^p$ according to the formula $\theta_j = \xi_j S_j$, $j = 1, \ldots, p$, where $S_j$ is the $j$th line of the matrix S. The offset values $\left\{\theta_j^0\right\}_{j=1}^p$ are defined in the same way as in [11], i. e., according to the uniform grid on the interval $[-r, r]$.

In [14], written by the present authors, comparison of popular random initialization methods with the proposed method and several specially developed deterministic initialization methods can be found.

## 3.3 Experimental Results

The proposed algorithms for the initialization step should be compared with the standard approaches in terms of the average error of approximation and its variations. For generation of test problems, a set of artificial multidimensional functions has been used for testing optimization algorithms. This choice is due to the fact that in the framework of surrogate modeling, constructed approximation is often used as the objective function instead of original function. Let distinguish two types of test problems:

(1) test functions with fixed dimension of the input vector $\vec{x}$ — allinit, beale, hartmann, ishigami, wbd; and

(2) test functions with dimension of the input vector $\vec{x}$ that can be set to some predefined value — gSobol, michalewicz, rosenbrock, whitley, zdt3. In experiments, the input dimension was varied from 3 to 10.

Exact formulas for the test functions can be found in [15, 16].

As a relative approximation error, the root-mean-square error normed by the analogous error for the constant approximation was used:

$$E\left(S_{\text{test}}, \hat{f}\right) = \sqrt{\frac{\sum_{i=1}^{N_{\text{test}}} \left(y_i - \hat{f}\left(\vec{x}_i\right)\right)^2}{\sum_{i=1}^{N_{\text{test}}} \left(y_i - \bar{y}\right)^2}} \qquad (4)$$

where

$$\bar{y} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} y_i \,.$$

The error comparable with 1 corresponds to a very inaccurate approximation and the error comparable with $10^{-3}$ corresponds to a very accurate approximation. In order to calculate the error, a separate test set $S_{\text{test}}$ has been used.

For each function (or pair "function-input dimension" for test problems of the second kind), the learning sample size was selected such that the approximation error (4), obtained using a standard approximation method, belonged to the interval 0.01–0.2 since this range is the most interesting for practice (usually, smaller values are not required in practice). As a standard approximation method, the realization of ANN from MatLab such that NW algorithm is used for the initialization, the validation sample is extracted randomly, and the training is performed by the Levenberg–Marquardt algorithm. In order to select the optimal dictionary

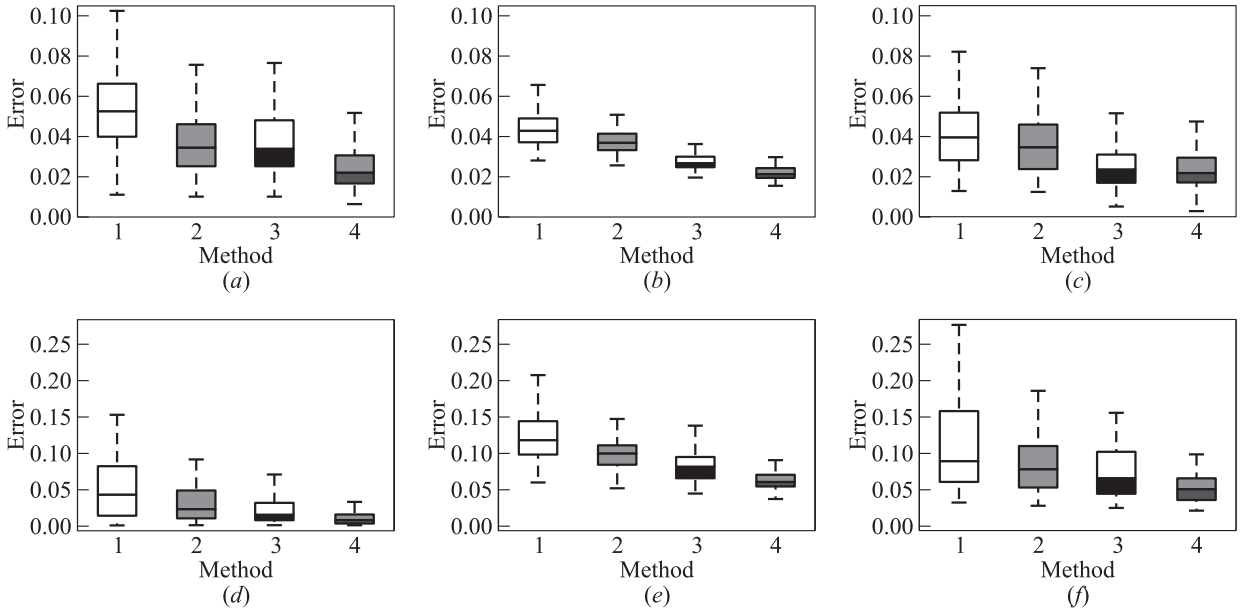



**Figure 1** Methods for solving problems of the initialization step (errors for some tests): (*a*) Whitley, $d = 3$, $N = 350$; (*b*) gSobol, $d = 3$, $N = 500$; (*c*) wbd, $d = 3$, $N = 75$; (*d*) Rosenbrock, $d = 3$, $N = 350$; (*e*) Hartmann, $d = 3$, $N = 500$; and (*f*) Allinit, $d = 3$, $N = 75$

size (the number of functions $p$) in each experiment, the brute force algorithm has been used with the criterion defined by the approximation error, estimated using cross-validation. Each separate experiment (the test function, the input dimension, and the learning sample size are fixed) has the following setup: 10 random learning sample $S_{\text{learning}}$ are generated; for each sample, 10 approximators are constructed. Let compare the following algorithms:

- the standard ANN algorithm (the methods are denoted on plots by number 1);
- the standard ANN with the proposed method for the validation sample extraction (method 2);
- the standard ANN with the proposed method for parameters initialization (method 3); and
- the standard ANN with both proposed methods for the initialization step (method 4).

In most cases, the proposed algorithms for solution the problems of the initialization step improve the final quality of the approximation (applied either independently or simultaneously). It is interesting that combination of the proposed algorithms (method 4) in most cases makes it possible to obtain the model with the superior accuracy compared with the accuracy of the model obtained using these algorithms independently. Therefore, new algorithm is more effective, see examples of diagrams in Fig. 1 and also, Dolan–More curves in section 5.

# 4 Approximation Training

## 4.1 Separability of Variables

Let consider the error function on the training sample as a function of the parameters $\Theta$ and $\vec{\alpha}$:

$$Q\left(S_{\text{train}}, \hat{f}\right) = Q\left(\Theta, \vec{\alpha}\right) = \left(\vec{y} - \hat{f}\left(X, \Theta, \vec{\alpha}\right)\right)^{\mathrm{T}} \left(\vec{y} - \hat{f}\left(X, \Theta, \vec{\alpha}\right)\right).$$

Note that the dependence of the error function $Q$ on the dictionary functions parameters $\Theta$ is nonlinear and very complex. At the same time, the dependence of $Q$ on expansion coefficients $\vec{\alpha}$ is quadratic. For the fixed $\Theta$, the optimal values of $\vec{\alpha}$ can be found by the least squares method:

$$\vec{\alpha}\left(\Theta\right) = \left(\Psi\left(\Theta\right)^{\mathrm{T}} \Psi\left(\Theta\right)\right)^{-1} \Psi\left(\Theta\right)^{\mathrm{T}} \vec{y} \qquad (5)$$

where

$$\Psi\left(\Theta\right) = \left\{\vec{\psi}\left(\Theta, \vec{x}_i\right), i = 1, \ldots, N_{\text{train}}\right\}.$$

Let take this fact into account when tuning parameters of the approximator. For this, let consider the objective function $R\left(\Theta\right) = Q\left(\Theta, \vec{\alpha}\left(\Theta\right)\right)$ where $\vec{\alpha}\left(\Theta\right)$ is calculated according to (5). When calculating the expansion coefficient according to (5), a nonlinear dependence $\vec{\alpha} = \left(\Theta\right)$ should be taken into account when

calculating the derivatives $R_\Theta$ and $R_{\Theta\Theta}$. An algorithm for such calculations was proposed in [17] and its theoretical properties were investigated in [18].

However, this algorithm has significant shortcoming: in (5), inversion of matrix $\Psi^{\mathrm{T}}\Psi$ which, in general, can be degenerated, was used. In such case, the inverse matrix does not exist and it is not reasonable to use the error function $R(\Theta)$. This situation was investigated in details in the framework of the linear regression methods [19]. It can be shown that even if the matrix $\Psi^{\mathrm{T}}\Psi$ does not degenerate but is ill-conditioned, then estimates of the coefficients $\vec{\alpha}$ are unstable (in statistical terms, this means that the estimate of $\vec{\alpha}$ has very big variance). In such case, the calculated values of the gradient and the Hessian of the error function are also unstable: even with small variations of the parameters $\Theta$, the first and the second derivatives of the error function change significantly resulting in a very low training speed.

Let consider a classical approach for regularization in linear regression problems, namely, ridge regression [19]. Let add to the matrix $\Psi^{\mathrm{T}}\Psi$ a positively definite matrix $\lambda I_p$, where $I_p$ is a unit matrix with sizes $p \times p$ and $\lambda > 0$ is a some constant. In such case, formula (5) takes the form

$$\vec{\alpha}(\Theta) = \left(\Psi(\Theta)^{\mathrm{T}}\Psi(\Theta) + \lambda I_p\right)^{-1}\Psi(\Theta)^{\mathrm{T}}\vec{y}. \quad (6)$$

It is obvious that any level of matrix $\left(\Psi(\Theta)^{\mathrm{T}}\Psi(\Theta) + \lambda I_p\right)$ conditionality can be reached by increasing the value of $\lambda$. Use of the ridge regression is equivalent to redefinition of the error function in the following way:

$$\begin{aligned}\widetilde{R}(\Theta) &= \widetilde{Q}(\Theta, \vec{\alpha}(\Theta)) \\ &= \left(\vec{y} - \hat{f}(X, \Theta, \vec{\alpha}(\Theta))\right)^{\mathrm{T}}\left(\vec{y} - \hat{f}(X, \Theta, \vec{\alpha}(\Theta))\right) \\ &\quad + \lambda\vec{\alpha}(\Theta)^{\mathrm{T}}\vec{\alpha}(\Theta)\,.\end{aligned}$$

**Statement 1.** *The gradient and the Hessian of the error function $\widetilde{R}$ can be calculated according to the following formulas*:

$$\begin{aligned}\widetilde{R}_\Theta &= \widetilde{Q}_\Theta = \vec{e}^{\mathrm{T}}\mathrm{J}\,; \\ \widetilde{R}_{\Theta\Theta} &= \mathrm{J}^{\mathrm{T}}\mathrm{J} + \vec{e}\odot\hat{f}_{\Theta\Theta} - \left(\vec{e}^{\mathrm{T}}\odot\Psi_\Theta + \mathrm{J}^{\mathrm{T}}\Psi\right) \\ &\quad \times\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}\right)^{-1}\left(\vec{e}^{\mathrm{T}}\odot\Psi_\Theta + \mathrm{J}^{\mathrm{T}}\Psi\right)^{\mathrm{T}}. \quad (7)\end{aligned}$$

Proof. Let find the first derivatives of the function $\widetilde{R}(\Theta)$:

$$\widetilde{R}_\Theta = \widetilde{Q}_\Theta + \widetilde{Q}_\alpha\alpha_\Theta = \widetilde{Q}_\Theta + \vec{0}\alpha_\Theta = \widetilde{Q}_\Theta\,. \quad (8)$$

Since coefficients $\vec{\alpha}(\Theta)$, obtained using (6), are the minimizer of the function $\widetilde{Q}(\Theta, \vec{\alpha}(\Theta))$, $\widetilde{Q}_\alpha = \vec{0}$. Let differentiate equality (8) with respect to $\Theta$:

$$\widetilde{R}_{\Theta\Theta} = \widetilde{Q}_{\Theta\Theta} + \widetilde{Q}_{\Theta\alpha}\alpha_\Theta\,. \quad (9)$$

Contrary to (8), the second summand is not equal to 0. Therefore, it is necessary to calculate $\alpha_\Theta$. Taking into account the equality $\widetilde{Q}_\alpha = \vec{0}$ as a consequence, one gets that

$$d\widetilde{Q}_\alpha = \widetilde{Q}_{\alpha\alpha}d\alpha + \widetilde{Q}_{\alpha\Theta}d\Theta = 0\,.$$

Consequently,

$$\alpha_\Theta = \frac{d\alpha}{d\Theta} = -\widetilde{Q}_{\alpha\alpha}^{-1}\widetilde{Q}_{\alpha\Theta}\,.$$

Then formula (9) takes the form

$$\widetilde{R}_{\Theta\Theta} = \widetilde{Q}_{\Theta\Theta} - \widetilde{Q}_{\Theta\alpha}\widetilde{Q}_{\alpha\alpha}^{-1}\widetilde{Q}_{\alpha\Theta}\,.$$

Now let write explicitly expressions for the derivatives of the function $\widetilde{Q}$. Let denote by $\mathrm{J} = \hat{f}_\Theta(X)$ the matrix of the derivatives of the model $\hat{f}$ with respect to $\Theta$ at the points $S_{\mathrm{train}}$. Therefore,

$$\left.\begin{aligned}\widetilde{Q}_\Theta &= \vec{e}^{\mathrm{T}}\mathrm{J}\,; \\ \widetilde{Q}_{\Theta\Theta} &= \mathrm{J}^{\mathrm{T}}\mathrm{J} + \vec{e}\odot\hat{f}_{\Theta\Theta}\,; \\ \widetilde{Q}_{\alpha\alpha} &= \Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}\,; \\ \widetilde{Q}_{\Theta\alpha} &= \vec{e}^{\mathrm{T}}\odot\Psi_\Theta + \mathrm{J}^{\mathrm{T}}\Psi\,.\end{aligned}\right\} \quad (10)$$

Final formula (7) can be obtained directly from

$$\widetilde{R}_{\Theta\Theta} = \widetilde{Q}_{\Theta\Theta} - \widetilde{Q}_{\Theta\alpha}\widetilde{Q}_{\alpha\alpha}^{-1}\widetilde{Q}_{\alpha\Theta}$$

using expressions for the derivatives from (10).

Assuming the residual vector $\vec{e}$ to be small, all terms with $\vec{e}$ in (7) can be neglected:

$$\widetilde{R}_{\Theta\Theta} \approx \mathrm{J}^{\mathrm{T}}\mathrm{J} - \left(\mathrm{J}^{\mathrm{T}}\Psi\right)\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}\right)^{-1}\left(\mathrm{J}^{\mathrm{T}}\Psi\right)^{\mathrm{T}}. \quad (11)$$

Therefore, one gets explicit formulas for calculation of the gradient and the approximate Hessian matrix of the modified error function $\widetilde{R}(\Theta)$ that are necessary for optimization based on the Levenberg–Marquardt algorithm (see formulas (1) and (2)).

Let note that the product $\mathrm{J}^{\mathrm{T}}\mathrm{J} \approx \widetilde{Q}_{\Theta\Theta}$ is nonnegatively definite. Let show that the Hessian $\widetilde{R}_{\Theta\Theta}$ also has this property. In such case, one can construct accurate local-quadratic approximations of the error function during its optimization.

**Statement 2.** *The matrix*

$$\mathrm{H} \stackrel{\mathrm{def}}{=} \mathrm{J}^{\mathrm{T}}\mathrm{J} - \left(\mathrm{J}^{\mathrm{T}}\Psi\right)\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}\right)^{-1}\left(\mathrm{J}^{\mathrm{T}}\Psi\right)^{\mathrm{T}}$$

*is nonnegatively definite for any* $\lambda \geq 0$.

Proof. Let rewrite matrix H in the following form:

$$\mathrm{H} = \mathrm{J}^{\mathrm{T}}\mathrm{J} - \left(\mathrm{J}^{\mathrm{T}}\Psi\right)\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}_p\right)^{-1}\left(\mathrm{J}^{\mathrm{T}}\Psi\right)^{\mathrm{T}} = \mathrm{J}^{\mathrm{T}}\left(\mathrm{I}_N - \Psi\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}_p\right)^{-1}\Psi^{\mathrm{T}}\right)\mathrm{J}\,.$$

Let $\Psi = \mathrm{VQU}^{\mathrm{T}}$ be a singular value decomposition for the matrix of regressors, then

$$\Psi\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}_p\right)^{-1}\Psi^{\mathrm{T}} = \mathrm{VQU}^{\mathrm{T}}\left(\mathrm{UQ}^2\mathrm{U}^{\mathrm{T}} + \lambda\mathrm{UU}^{\mathrm{T}}\right)^{-1}\mathrm{UQV}^{\mathrm{T}} = \mathrm{VQU}^{\mathrm{T}}\mathrm{U}\left(\mathrm{Q}^2 + \lambda\mathrm{I}_p\right)^{-1}\mathrm{U}^{\mathrm{T}}\mathrm{UQV}^{\mathrm{T}} = \mathrm{VQ}^2\left(\mathrm{Q}^2 + \lambda\mathrm{I}_p\right)^{-1}\mathrm{V}^{\mathrm{T}}\,.$$

Let $\tilde{q}_j$ be eigenvalues of the matrix $\mathrm{VQ}^2\left(\mathrm{Q}^2 + \lambda\mathrm{I}_p\right)^{-1}\mathrm{V}^{\mathrm{T}}$, then $\tilde{q}_j = q_j^2/(q_j^2 + \lambda)$ where $q_j$ are the elements of the matrix $Q$. The eigenvalues of the matrix $\mathrm{P} = \mathrm{I}_N - \Psi\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}_p\right)^{-1}\Psi^{\mathrm{T}}$ are not smaller than the difference of the minimal eigenvalue of $\mathrm{I}_N$ and the maximal eigenvalue from the set $\tilde{q}_j$. It is obvious that this difference is nonnegative since

$$1 - \max_j \tilde{q}_j = 1 - \max_j\left(\frac{q_j^2}{q_j^2 + \lambda}\right) \geq 0$$

for $\lambda \geq 0$. Therefore, the matrix P is nonnegatively definite and the matrix $\mathrm{H} = \mathrm{J}^{\mathrm{T}}\mathrm{PJ}$ is also nonnegatively definite.

Now let estimate the computational complexity of formula (11) for the Hessian calculation $\widetilde{R}_{\Theta\Theta}$. The size of the Jacobian matrix J is equal to $N_{\text{train}} \times p(d+1)$, the size of the matrix with regressors $\Psi$ is equal to $N_{\text{train}} \times p$. It is necessary to perform $O(N_{\text{train}}p^2d^2)$ operations in order to calculate the main term $\mathrm{J}^{\mathrm{T}}\mathrm{J}$, which is needed also for calculation of the standard error function. At the same time, additional summand $\left(\mathrm{J}^{\mathrm{T}}\Psi\right)\left(\Psi^{\mathrm{T}}\Psi + \lambda\mathrm{I}\right)^{-1}\left(\mathrm{J}^{\mathrm{T}}\Psi\right)^{\mathrm{T}}$ can be calculated for

$$O(N_{\text{train}}p^2d + p^3 + p^3d + p^3d^2) = O(N_{\text{train}}p^2d + p^3d^2)$$

operations. Since in the considered class of approximation problems $N_{\text{train}} \gg d$, $N_{\text{train}} \gg p$, then calculation of the additional summand requires significantly smaller number of operations compared to the number of operations for calculation of the main summand.

The proposed error function $\widetilde{R}(\Theta)$ does not only increase approximation accuracy but also decreases training time compared to the training time when using the standard error function $Q(\Theta, \vec{\alpha})$ [20]. This advantage can be explained by two reasons. Some part of the parameters are estimated optimally on each iteration of the training algorithm and adaptive regularization increases numerical stability of the training process. In this work, it is assumed that the output $y$ dimension is equal to 1, but for many applied problems, the output $y$ can be multidimensional and its dimension $d_y$ can even be higher than the input dimension $d$. In such case, the number of parameters of the standard error function is $(d + d_y)/d$ times higher than that of the modified error function. Thus, separability of the variables can significantly decrease the number of optimized parameters in some problems.

## 4.2 Adaptive Regularization

Standard approaches for regularization of models with the structure $\hat{f}(\vec{x}) = \vec{\psi}(\Theta, \vec{x})\,\vec{\alpha}$ use the $L_2$ penalty on all parameters of the model [3] and a regularization coefficient is determined experimentally using some additionally extracted validation samples and multiple training of the surrogate model. In [21], some Bayesian approach is proposed for the regularization parameter selection and, again, the $L_2$ penalty on all parameters of the model are used. This method has two key shortcomings:

(1) selection of the regularization parameter does not take into account the error of approximation; and

(2) the penalty incorporates norms of the parameters $\Theta$ and $\vec{\alpha}$ with equal weights and does not take into account essentially different nature of these parameters.

In the proposed approach, only the expansion coefficients $\vec{\alpha}$ have been penalized and regularization parameter $\lambda$ has been selected optimally (in some sense) with taking into account the error of approximation.

When tuning the regularization parameter $\lambda$ during the training process, the functional dependency $\lambda = \lambda(\Theta)$ appears. In general case, one should take into account this dependency when calculating the derivatives of the error function with respect to $\Theta$. However, in the framework of the considered approximation problem, optimization of the error function on the set $S_{\text{train}}$ can be considered as the process for generation of different models with the final aim to obtain the model with the smallest error on the independent test set $S_{\text{test}}$ rather than for finding the minimum of the error function on the train set $S_{\text{train}}$. Due to this remark, the change of the regularization parameter value during the training process is allowed and these changes can be neglected when calculating the derivatives of the error function.

In order to select the regularization parameter $\lambda$ on each iteration of the training algorithm, let minimize the GCV (Generalized Cross Validation) criterion [3] estimating the approximation error on the test sample in linear regression problems ($\vec{\alpha}(\lambda)$ is calculated according to (6)):

$$GCV(\Psi,\lambda)=\frac{\sum\limits_{i=1}^{N_{\text{train}}}\left(y_i-\hat{f}(x_i)\right)^2}{\left(1-(1/N_{\text{train}})\text{tr}\left(\text{L}\right)\right)^2}$$
$$=\frac{\left(\vec{y}-\Psi\vec{\alpha}(\lambda)\right)^{\text{T}}\left(\vec{y}-\Psi\vec{\alpha}(\lambda)\right)}{\left(1-(1/N_{\text{train}})\text{tr}\left(\left(\vec{\Psi}^{\text{T}}\vec{\Psi}+\lambda\text{I}\right)^{-1}\vec{\Psi}^{\text{T}}\vec{\Psi}\right)\right)^2}.$$

Let note that when minimizing the criterion GCV with respect to $\lambda$ it is necessary to control condition number of the matrix $\Psi^{\text{T}}\Psi+\lambda I$ in order the training process to be stable [22]. It is proposed to impose a lower bound on the value of $\lambda$ such that provides necessary level of the conditionality ($10^{12}$ in the present realization of the algorithm).

### 4.3 Experimental Results

Let use the testing methodology, described in subsection 3.3 Let compare the following methods: the standard method (method 1), the method incorporating algorithms from subsections 3.1 and 3.2 (method 4), the method incorporating only the modified error function (method 5), and the method incorporating all the proposed algorithms (method 6).

The most indicative results are given in Fig. 2. The proposed modification of the error function allows to significantly improve approximation quality for some problems (for example, function michalewicz). If method 6, which incorporates all the proposed algorithms, is considered, then one can see that this method does not only additionally increase approximation accuracy compared to the best of two methods 4 and 5, but also significantly decreases the approximation error in some cases (for example, zdt3 function).

## 5 Experimental Results

In this section, the results of full experiments on artificial functions will be shown and the proposed approach will be compared with other similar approaches on some applied problems of surrogate modeling. Let use Dolan–More curves $P_k(a)$ [23] for visualization of the results. The quantity $P_k(a)$ shows on which fraction of the problems the errors of the considered approximation method $k$ are not $a \geq 1$ higher than the minimal (among all considered methods) approximation error for the corresponding approximation problems.

### 5.1 Artificial Functions

Using Dolan–More curves, let compare the standard algorithm with all algorithms from subsections 3.3 and 4.3 on all problems used for testing. As a "separate problem," let consider all independently generated samples, i. e., 10 problems correspond to each function. Let use the following criteria of approximation quality: median of the errors (accuracy) and standard deviation of the

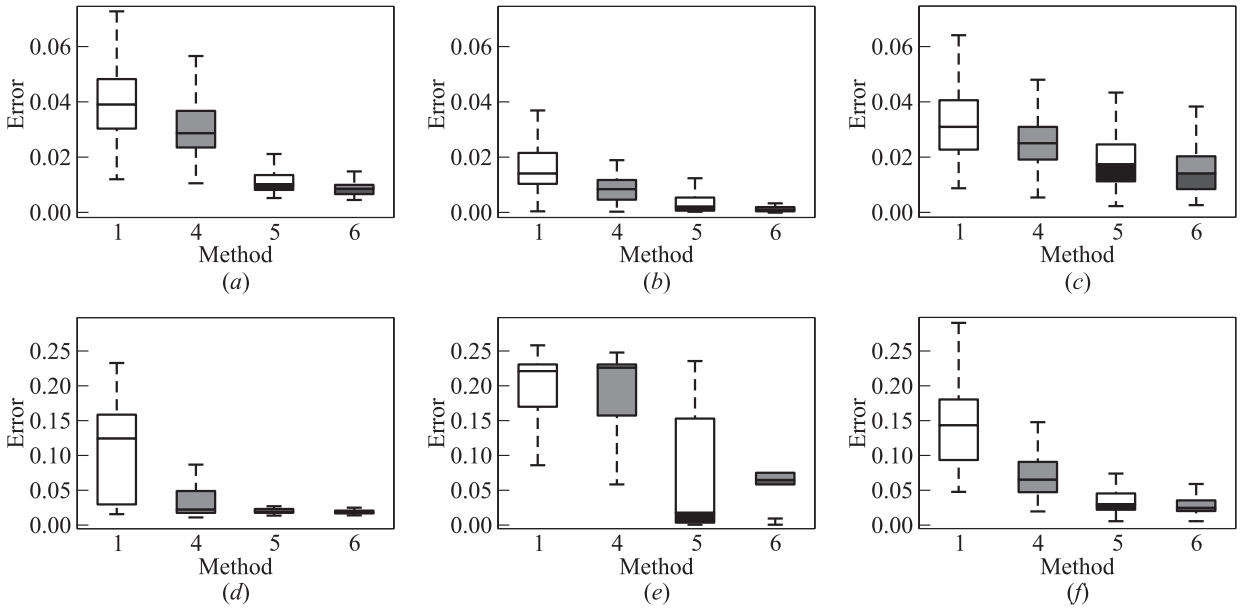



**Figure 2** Methods for solving problems of the training step (errors for some tests): (*a*) Whitley, $d=4$, $N=750$; (*b*) wbd, $d=4$, $N=125$; (*c*) Beale, $d=3$, $N=100$; (*d*) Michalewicz, $d=5$, $N=250$; (*e*) zdt, $d=3$, $N=500$; and (*f*) Ishigami, $d=3$, $N=250$

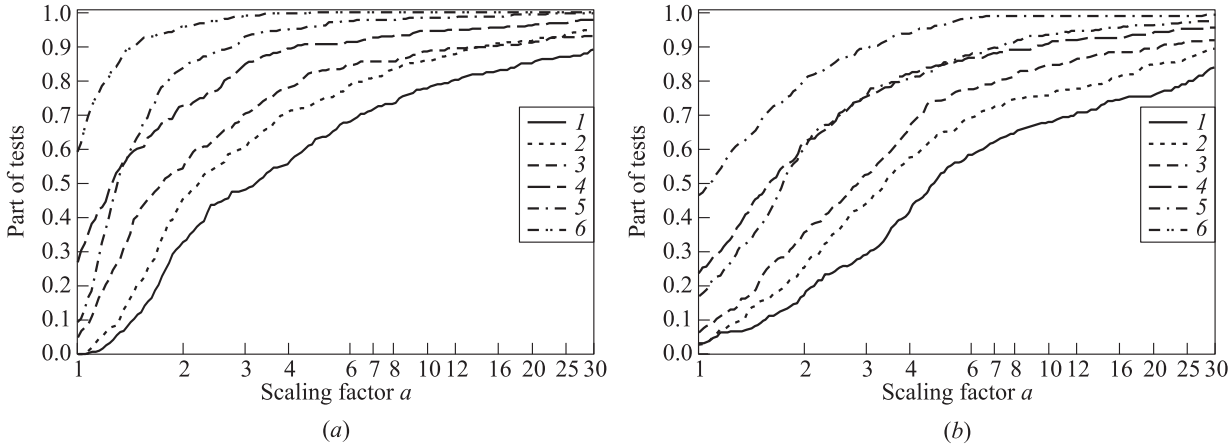



**Figure 3** Results for artificial problems: (*a*) median values (accuracy) and (*b*) standard deviation (scatter) (*1* — standard algorithm; *2* — extraction of Svalidation; *3* — initialization of $\theta$; *4* — extraction + initialization; *5* — training algorithm; and *6* — all proposed algorithms)

errors (scatter of the errors). For each problem, let run each method 10 times (the error for each run is estimated using formula (4)) and estimate the accuracy and the scatter using results of these runs.

The results of comparison are given in Fig. 3. One can see that the main contribution into accuracy is obtained due to the modified error function with the adaptive regularization, but the algorithms of the initialization step also significantly improve the accuracy of the methods (see curves for the methods 5 and 6).

## 5.2 Real Applied Problems

Let compare approximation quality of the proposed method (combining all the proposed algorithms) with widely-used methods for approximation construction. There was used realization of such methods in MatLab and modeFrontier. This software systems are used by many of industrial companies. There are a number of methods for approximation construction, implemented in MatLab and modeFrontier, namely, ANN, regression based on Gaussian Process, etc. Note that many of these methods have serious restrictions on possible characteristics of the sample (input dimension $d$ and sample size $N_{\text{learning}}$) which, in turn, limits the applicability of the methods and, as a consequence, reduces the accuracy of approximation. In most of the cases, such restrictions are due to algorithmic peculiarities of the corresponding realizations.

Let consider results on some indicative problems covering a wide range of sample sizes and input dimensions:

– the strength of the composite structure of the aircraft fuselage [4];

**Table 1** Relative approximation error

| Problem | | Composite structure | Wing characteristics | Sand | Concrete |
|---|---|---|---|---|---|
| Dimension of $\vec{x}$ | | 16 | 78 | 3 | 8 |
| Sample size | | 50000 | 65000 | 10000 | 1030 |
| Proposed approach | | 0.092 | 0.159 | 0.091 | 0.320 |
| MatLab | Linear regression | 0.616 | 0.330 | 0.698 | 0.6391 |
| | Quadratic regression | 0.390 | — | 0.608 | 0.485 |
| | ANN | 0.194 | 0.258 | 0.132 | 0.350 |
| | Radial basis function (RBF) | 0.336 | 0.628 | 0.464 | 0.371 |
| mode Frontier | $k$-nearest neighbors | 0.597 | 1.115 | 0.470 | 0.529 |
| | Anisotronic kriging | 0.528 | — | 0.813 | 2.521 |
| | Kriging | 0.382 | 1.031 | 0.937 | 0.889 |
| | RBF | 0.363 | 9.392 | 0.494 | 0.367 |
| | ANN | 0.299 | 1.424 | 0.356 | 0.730 |
| | Gaussian process | 0.807 | — | 0.561 | 2.088 |

– aerodynamic characteristics of the aircraft wing [24];
– structure of the sand in the field [25]; and
– concrete compressive strength [26].

Reference results (of approximation algorithms from MatLab and modeFrontier) were obtained in 2010 during the work on the PhD thesis. The approximation quality was measured using error (4), calculated using the independent test set $S_{\text{test}}$. The results of the comparison are given in Table 1 (few methods do not work for the problem of approximation of the aircraft wing aerodynamic characteristics due to high input dimensionality).

# 6 Concluding Remarks

The problem of approximation of a multidimensional dependency has been considered based on a linear expansion in a dictionary of parametric functions. The new methods have been proposed for solving subproblems that arise in the framework of approximation construction problem. Each of these methods as well as their combination significantly increases the approximation quality compared to the approximation quality obtained using standard methods. Using the proposed algorithm, it was possible to solve important applied problems, (see, for example, [4]).

# References


1. *Forrester A., Sobester A., Keane A.* Engineering design via surrogate modelling. A practical guide. — Wiley, 2008.
2. *Kuleshov A., Bernstein A.* Cognitive technologies in adaptive models of complex plants // Keynote Papers of 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09), 2009. P. 70–81.
3. *Hastie T., Tibshirani R., Friedman J.* The elements of statistical learning: Data mining, inference, and prediction. — Springer, 2008.
4. *Grihon S., Alestra S., Burnaev E., Prikhodko P.* Optimization of composite structure based on surrogate modelling of buckling analysis // Information Technologies and Systems Conference Proceedings, 2012. P. 41–47.
5. *Petrushev P.* Approximation by ridge functions and neural networks // SIAM J. Math. Anal. 30, 1998. P. 155–189.
6. *Pinkus A.* Approximation theory of the MLP model in neural networks // Acta Numerica, 1999. Vol. 8. P. 143–195.
7. *Vapnik V. N., Chervonenkis A. Ja.* Ordered risk minimization (I and II) // Autom. Remote Control, 1974. Vol. 34. P. 1226–1235; 1403–1412.
8. *Nocedal J., Wright S.* Numerical optimization. — 2nd ed. — Springer, 2006. P. 664.
9. *Marquardt D. W.* An algorithm for least-squares estimation of nonlinear parameters // J. SIAM, 1963. Vol. 11. No. 2. P. 431–441.
10. *Breiman L.* Classification and regression trees. — Wadsworth, 1984.
11. *Nguyen D., Widrow B.* Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights // IJCNN Joint Conference (International), 1990. P. 21–26.
12. *Drago G., Ridella S.* Statistically controlled activation weight initialization (SCAWI) // Trans. Neur. Netw., IEEE Press, 1992. Vol. 3. No. 4. P. 627–631.
13. *Rubinstein R. Y.* Generating random vectors uniformly distributed inside and on the surface of different regions // Eur. J. Oper. Res., 1982. Vol. 10. No. 2. P. 205–209.
14. *Belyaev M. G., Burnaev E. V., Erofeev P. D., Prikhodko P. V.* Comparison of the efficiency of the initialization methods for non-linear regression models // Information Technologies and Systems Conference Proceedings, 2011. P. 315–320.
15. *Hedar A.-R.* Global optimization test problems // http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.
16. *Molga M., Smutnicki C.* Test functions for optimization needs // www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.
17. *Golub G. H., Pereyra V.* The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate // SIAM J. Numer. Anal., 1973. Vol. 10. P. 413–432.
18. *Ruhe A., Wedin P. A.* Algorithms for separable nonlinear least squares problems // SIAM Review, 1980. Vol. 22. No. 3. P. 318–337.
19. *Demidenko E. Z.* Linear and non-linear regression. — Finance and stochastics. 1981.
20. *Belyaev M. G., Lyubin A. D.* Peculiarities of the optimization problem, which arises when constructing approximation of multidimensional function // Information Technologies and Systems Conference Proceedings, 2011. P. 415–422.
21. *Foresee D., Hagan M.* Gauss-Newton approximation to Bayesian learning // Conference (International) on Neural Networks Proceedings, 1997. Vol. 3. P. 1930–1935.
22. *Belyaev M. G., Burnaev E. V.* Adaptive regularization in the problem of multidimensional functions approximation // Information Technologies and Systems Conference Proceedings, 2009. P. 431–435.
23. *Dolan E., More J.* Benchmarking optimization software with performance profiles // Math. Programming, Ser. A, 2002. Vol. 91. P. 201–213.
24. *Chervonenkis A. Ya., Chernova S. S., Zykova T. V.* Applications of kernel ridge estimation to the problem of computing the aerodynamical characteristics of a passenger plane (in comparison with results obtained with artificial neural networks) // Automation Remote Control, 2011. Vol. 72. Iss. 5. P. 1061-1067.
25. IC Fault dataset. imperial.ac.uk/earthscienceandengineering/research/perm/icfaultmodel.
26. Concrete Compressive Strength dataset. archive.ics.uci.edu/ml/datasets.

# АППРОКСИМАЦИЯ МНОГОМЕРНЫХ ЗАВИСИМОСТЕЙ НА ОСНОВЕ РАЗЛОЖЕНИЯ ПО СЛОВАРЮ ПАРАМЕТРИЧЕСКИХ ФУНКЦИЙ

М. Г. Беляев[1], Е. В. Бурнаев[2]

[1] Институт проблем передачи информации РАН, Московский физико-технический институт (государственный университет); ООО Датадванс, belyaev@iitp.ru

[2] Институт проблем передачи информации РАН, Московский физико-технический институт (государственный университет); ООО Датадванс, burnaev@iitp.ru

**Аннотация:** Рассматривается задача аппроксимации многомерной зависимости по конечному множеству пар «точка» – «значение функции в точке». Для решения этой задачи используется модель зависимости, представляющая собой разложение по словарю нелинейных параметрических функций. Построение аппроксимации, основанной на этой модели, может быть разбито на несколько подзадач: выделение валидационной подвыборки, инициализация параметров функций словаря, последующая настройка параметров функций словаря. Предложены эффективные методы решения этих подзадач. Описанный подход демонстрирует высокое качество работы на ряде задач инженерного проектирования и успешно применяется в реальных приложениях.

**Ключевые слова:** нелинейная аппроксимация; словарь параметрических функций

# ABSTRACTS

## UNSUPERVISED APPROACH TO WEB WRAPPER MAINTENANCE

A. M. Andreev[1], D. V. Berezkin[2], I. A. Kozlov[3], and K. V. Simakov[4]

[1]Bauman Moscow State Technical University, arkandreev@gmail.com
[2]Bauman Moscow State Technical University, dmitryb2007@yandex.ru
[3]Bauman Moscow State Technical University, kozlovilya89@gmail.com
[4]Bauman Moscow State Technical University, skv@ixlab.ru

HTML-wrapper applications rely on formatting regularities of targeted websites. Therefore, maintenance of such applications is connected with the problem of detecting markup changes of web pages. This article describes the unsupervised approach to this problem. The proposed method of detection consists of two parts: the real-time one based on clustering considering HTML-document as a vector of some features and the time-lagged one based on comparison of distributions of such features for learning and testing sets of HTML-documents. There have been carried out several experiments with data obtained from real wrapper. The results reveal feasibility of the suggested approach.

**Keywords:** wrapper maintenance; web-site parsing; clustering; HTML-markup statistical processing

## BUILDING REAL-TIME NEWS RECOMMENDATION SERVICE USING NoSQL DBMS

P. A. Klemenkov

M. V. Lomonosov Moscow State University, parser@cs.msu.su

The analysis of user interaction with a Web application, the methods of conducting such an analysis, and their shortcomings are discussed. An implementation of the news recommendation service using existing approaches is described. A new NoSQL approach to building recommendation systems that operate in near real time is suggested.

**Keywords:** recommendation systems; minhash; mapreduce; NoSQL

## A VERIFIABLE MAPPING OF A MULTIDIMENSIONAL ARRAY DATA MODEL INTO AN OBJECT DATA MODEL

S. A. Stupnikov

IPI RAN, ssa@ipi.ac.ru

The paper considers a mapping of a multidimensional array data model into an object data model. General principles of mappings of array data models into object data models are formulated. A mapping of concrete models is also considered. The source model is the Array Data Model used in the SciDB DBMS. The target model is the SYNTHESIS language used as the canonical data model in the subject mediation technology. A method for verification of the mapping is considered. Verification means a formal proof that the mapping preserves information and semantics of the operations. Verification is realized using the AMN formal specification language. A practical aim of the paper is to provide a basis for virtual or materialized integration of array-based information resources.

**Keywords:** multidimensional arrays; object data model; data model mapping; database integration

## STUDY OF THE WIKIPEDIA(EN) CATEGORIES GRAPH

A. V. Shkotin

GIS department, State Geological Museum of the Russian Academy of Sciences, ashkotin@acm.org

Wikipedia is the outstanding project of knowledge accumulation both of general using and different areas of specialization. Quality check of this knowledge, especially automatic, is very important. In this paper, the results

of studying a structure of the English version of WCG (Wikipedia Categories Graph) as a whole are presented. The WCG is a system that supports structure of knowledge and it is interesting to know what WCG includes and how it is arranged. It is shown that in graph, there are unacceptable logic violations and organizational and technical methods for elimination are discussed.

**Keywords:** Wikipedia; digraph; connected components; logical analysis

## ACTIVE AUTHENTICATION METHODS USING KEYSTROKE DYNAMICS

V. Yu. Kaganov[1], A. K. Korolyov[2], M. N. Krylov[3], I. V. Mashechkin[4], and M. I. Petrovskiy[5]

[1] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, vladhid@mlab.cs.msu.su
[2] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, akorolev@mlab.cs.msu.su
[3] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, krylovm@mlab.cs.msu.su
[4] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, mash@cs.msu.su
[5] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University, michael@cs.msu.su

An overview of some effective methods of authentication using behavior models, created from keystroke dynamics data is presented. Also, a new data representation model was proposed, a number of experiments conducted using this model, and various algorithms of machine learing.

**Keywords:** wavelets; thresholding; risk estimate; normal distribution; rate of convergence

## PROBLEMS OF THE ONLINE ACCESS TO SCIENTIFIC JOURNALS

A. V. Glushanovskii[1] and N. E. Kalenov[2]

[1] Library for Natural Sciences, Russian Academy of Sciences, avglush@benran.ru
[2] Library for Natural Sciences, Russian Academy of Sciences, nek@benran.ru

The problems of supplying with full-text scientific information access via Internet for the institutions of the Russian Academy of Sciences (RAS) are considered. According to world practice, this task is resolved by the scientific libraries and libraries consortia for the best financial conditions. The practice of such access organization in Russia via Russian Foundation for Basic Research and National Electronic-information Consortia (NEICON) is described. The statistics of using NEICON provided online journals by RAS staff is considered. Organizational proposals for optimal decision of the task of online access to scientific information in the situation of financial limits in RAS are suggested.

**Keywords:** scientific journals; full texts; Internet; remote access; libraries; consortia

## DECISION SUPPORT SYSTEMS MODELING WITH SYNERGETIC ARTIFICIAL INTELLIGENCE

I. A. Kirikov[1], A. V. Kolesnikov[2], and S. V. Listopad[3]

[1] Immanuel Kant Baltic Federal University (в статье другое); Kaliningrad Branch of Institute of Informatics Problems, Russian Academy of Sciences, baltbipiran@mail.ru
[2] Kaliningrad Branch of Institute of Informatics Problems, Russian Academy of Sciences, avkolesnikov@yandex.ru
[3] Kaliningrad Branch of Institute of Informatics Problems, Russian Academy of Sciences, ser-list-post@yandex.ru

The approach to modeling collective effects of decision support systems within the paradigm of synergetic artificial intelligence is considered. The model and the functional structure of the hybrid intelligent multiagent system for

modeling decision support systems are proposed. The results of computational experiments that demonstrate a positive impact of the self-organization effect on the quality of collective decisions are presented.

**Keywords:** decision support computer system; hybrid intelligent multiagent system with self-organization

## SEMANTICS OF ASPECT-ORIENTED MODELING OF DATA AND PROCESSES

S. P. Kovalyov

Institute of Control Problems, Russian Academy of Sciences, kovalyov@nm.ru

An approach to semantic unification of aspect-oriented programming (AOP) technologies based on formalization by means of category theory is presented. Aspect-oriented programming technology is represented as a category of formal models of aspect-oriented programs and their interconnections equipped with functor of taking aspectual structure (labeling of models by concerns). Weaving of aspect-oriented programs is formalized as certain universal construction in this category. Formal AOP technologies applicable for reducing costs at modeling data and process scenarios are defined and considered. Weaving existence condition for scenario models is stated and justified.

**Keywords:** aspect-oriented programming; category theory; aspect weaving

## COGNITIVE INTEROPERABILITY OF EXPERT COLLABORATION IN THE TASK OF THE RUSSIAN-FRENCH PARALLEL TEXTS PROCESSING: LINGUISTIC AND COGNITIVE ASPECTS

O. S. Kozhunova

IPI RAN, kozhunovka@mail.ru

The resources of information and communication technologies “Refillable linguistic data base on translation difficulties” and “Subject-oriented thesaurus of Russian-French parallel texts” are discussed. The resources are at the design stage and to be implemented simultaneously with the Russian-French parallel corpus of belles-lettres. Apart from the functionality, linguistic and cognitive aspects of expert interaction within the task of the Russian-French parallel texts processing through cooperative efforts are considered.

**Keywords:** cognitive interoperability; task of natural language processing; Russian-French parallel texts

## DATA ACQUISITION SIMULATION FOR NICA EXPERIMENT

V. V. Korenkov[1], A. V. Nechaevskiy[2], and V. V. Trofimov[3]

[1]Joint Institute for Nuclear Research, Laboratory of Information Technologies Dubna, korenkov@cv.jinr.ru
[2]Joint Institute for Nuclear Research, Laboratory of Information Technologies Dubna, Andrey.Nechaevskiy@gmail.com
[3]Joint Institute for Nuclear Research, Laboratory of Information Technologies Dubna, trofimov@jinr.ru

The need for simulation model of data storage and processing for NICA accelerator complex is shown. The base of the simulation model is GridSim. This paper describes an approach to simulation the dCache and network. A simple example shows the case of the model use.

**Keywords:** grid technologies; grid infrastructures; data storage systems; optimization; simulation; research; development; dCache; Tier1; NICA; Grid

## ESTIMATES OF THE RATE OF CONVERGENCE OF THE DISTRIBUTIONS OF SOME RANDOM SUMS TO STABLE LAWS

V. Yu. Korolev[1] and L. M. Zaks[2]

[1] Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; IPI RAN, vkorolev@cs.msu.su

[2] Department of Modeling and Mathematical Statistics, Alpha-Bank, lily.zaks@gmail.com

Estimates are presented for the rate of convergence of the distributions of special sums of independent identically distributed random variables with finite variances to symmetric strictly stable laws. The distribution of the random index is assumed to be mixed Poisson in which the mixing distribution is a stable law concentrated on the positive half-line. The absolute constants are written out explicitly.

**Keywords:** stable distribution; Berry–Esseen inequality; random sum; doubly stochastic Poisson process (Cox process); mixed Poisson distribution

## UNIVERSAL METRIC THESAURUS OF RUSSIAN LANGUAGE

L. A. Kuznetsov[1], V. F. Kuznetsova, and A. V. Kapnin[3]

[1] Russian Presidential Academy of National Economy and Public Administration (Lipetsk Branch), kuznetsov.leonid48@gmail.com

[2] Russian Presidential Academy of National Economy and Public Administration (Lipetsk Branch), kuznetsov.leonid48@gmail.com

[3] Lipetsk State Technical University, gert@inbox.ru

All Russian language available thesauri are compiled by expert groups. In the paper, the tools for automatic generating of a thesaurus are presented. The tools are based on a formal presentation of the texts explaining semantics of the words and a quantify assessment of the semantic distance between the words as a measure of their proximity. The proposed solutions allow to use the formal mathematical presentations that minimize subjectivity in assessing the proximity of the words. The solutions give an opportunity to synthesize automatic systems for evaluating the semantic proximity of the words and to solve other problems in the area of texts processing.

**Keywords:** computational linguistics; universal thesaurus; metric thesaurus; semantic proximity assessment; semantic distance; information theory

## APPROXIMATION OF A MULTIDIMENSIONAL DEPENDENCY BASED ON LINEAR EXPANSION IN A DICTIONARY OF PARAMETRIC FUNCTIONS

M. G. Belyaev[1] and E. V. Bunaev[2]

[1] Institute for Information Transmission Problems RAS, Moscow Institute of Physics and Technology, Datadvance LLC, belyaev@iitp.ru

[2] Institute for Information Transmission Problems RAS, Moscow Institute of Physics and Technology, Datadvance LLC, burnaev@iitp.ru

The problem of a multidimensional function approximation using a finite set of pairs “point” – “function value at this point” is considered. As a model for the function, an expansion in a dictionary containing nonlinear parametric functions has been used. Several subproblems should be solved when constructing an approximation based on such model: extraction of a validation sample, initialization of parameters of the functions from the dictionary, and tuning of these parameters. Efficient methods for solving these subproblems have been suggested. Efficiency of the proposed approach is demonstrated on some problems of engineering design.

**Keywords:** nonlinear approximation; parametric dictionaries

# ОБ АВТОРАХ

**Андреев Аркадий Михайлович** (р. 1943) — кандидат технических наук, доцент МГТУ им. Н. Э. Баумана

**Беляев Михаил Геннадьевич** (р. 1987) — аспирант Института проблем передачи информации РАН; младший научный сотрудник Московского физико-технического института (государственного университета); научный сотрудник ООО «Датадванс»

**Березкин Дмитрий Валерьевич** (р. 1966) — кандидат технических наук, старший научный сотрудник МГТУ им. Н. Э. Баумана

**Бурнаев Евгений Владимирович** (р. 1983) — кандидат физико-математических наук, доцент; заведующий сектором Института проблем передачи информации РАН; старший научный сотрудник Московского физико-технического института (государственного университета); заведующий сектором ООО «Датадванс»

**Глушановский Алексей Валерианович** (р. 1944) — старший научный сотрудник Библиотеки по естественным наукам РАН

**Закс Лилия Михайловна** (р. 1989) — главный специалист отдела моделирования и математической статистики Альфа-банка

**Каганов Владислав Юрьевич** (р. 1993) — студент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

**Калёнов Николай Евгеньевич** (р. 1945) — доктор технических наук, профессор, директор Библиотеки по естественным наукам РАН

**Капнин Алексей Владимирович** (р. 1986) — аспирант, ассистент Липецкого государственного технического университета

**Кириков Игорь Александрович** (р. 1955) — кандидат технических наук, директор Калининградского филиала ИПИ РАН

**Клеменков Павел Андреевич** (р. 1986) — аспирант кафедры системного программирования факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

**Ковалёв Сергей Протасович** (р. 1972) — кандидат физико-математических наук, старший научный сотрудник Института проблема управления им. В. А. Трапезникова РАН

**Кожунова Ольга Сергеевна** (р. 1982) — кандидат технических наук, заведующая сектором ИПИ РАН

**Козлов Илья Андреевич** (р. 1989) — магистрант факультета информатики и систем управления МГТУ им. Н. Э. Баумана

**Колесников Александр Васильевич** (р. 1948) — доктор технических наук, профессор Балтийского федерального университета имени Иммануила Канта, старший научный сотрудник Калининградского филиала ИПИ РАН

**Кореньков Владимир Васильевич** (р. 1953) — кандидат физико-математических наук, директор лаборатории информационных технологий Объединенного института ядерных исследований (ОИЯИ); заведующий кафедрой Международного университета природы, общества и человека «Дубна»

**Королев Виктор Юрьевич** (р. 1954) — доктор физико-математических наук, профессор кафедры математической статистики факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова; ведущий научный сотрудник ИПИ РАН

**Королёв Андрей Константинович** (р. 1992) — студент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

**Крылов Михаил Николаевич** (р. 1992) — студент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

**Кузнецов Леонид Александрович** (р. 1942) — доктор технических наук, профессор, заведующий кафедрой Российской академии народного хозяйства и государственной службы при Президенте РФ (Липецкий филиал)

**Кузнецова Вера Федоровна** (р. 1948) — кандидат технических наук, доцент Российской академии народного хозяйства и государственной службы при Президенте РФ (Липецкий филиал)

**Листопад Сергей Викторович** (р. 1984) — кандидат технических наук, научный сотрудник Калининградского филиала ИПИ РАН

**Машечкин Игорь Валерьевич** (р. 1956) — доктор физико-математических наук, профессор факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

**Нечаевский Андрей Васильевич** (р. 1982) — инженер-программист лаборатории информационных технологий Объединенного института ядерных исследований (ОИЯИ)

**Петровский Михаил Игоревич** (р. 1975) — кандидат физико-математических наук, доцент факультета вычислительной математики и кибернетики Московского государственного университета им. М. В. Ломоносова

**Симаков Константин Васильевич** (р. 1980) — кандидат технических наук, старший научный сотрудник МГТУ им. Н. Э. Баумана

**Ступников Сергей Александрович** (р. 1978) — кандидат технических наук, старший научный сотрудник ИПИ РАН

**Трофимов Владимир Валентинович** (р. 1955) — ведущий программист лаборатории информационных технологий Объединенного института ядерных исследований (ОИЯИ)

**Шкотин Александр Владимирович** (р. 1952) — инженер-программист отдела ГИС Государственного геологического музея РАН

# ABOUT AUTHORS

**Andreev Arkady M.** (b. 1943) — Candidate of Science (PhD) in technology, assistant professor, Bauman Moscow State Technical University

**Belyaev Mikhail G.** (b. 1987) — PhD student, Institute for Information Transmission Problems, Russian Academy of Sciences; junior scientist, Moscow Institute of Physics and Technology; scientist, Datadvance LLC

**Berezkin Dmitry V.** (b. 1966) — Candidate of Science (PhD) in technology, senior scientist, Bauman Moscow State Technical University

**Burnaev Evgeny V.** (b. 1983) — Candidate of Science (PhD) in physics and mathematics, associate professor; Head of Laboratory, Institute for Information Transmission Problems, Russian Academy of Sciences; senior scientist, Moscow Institute of Physics and Technology; Head of Laboratory, Datadvance LLC

**Glushanovskiy Alexey V.** (b. 1944) — senior scientist, Library for Natural Sciences, Russian Academy of Sciences

**Kaganov Vladislav Yu.** (b. 1993) — student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

**Kalenov Nikolay E.** (b. 1945) — Doctor of Science in technology, professor, Director, Library for Natural Sciences, Russian Academy of Sciences

**Kapnin Alexey V.** (b. 1986) — PhD student, assistant professor of Lipetsk State Technical University

**Kirikov Igor A.** (b. 1955) — Candidate of Science (PhD) in technology, Director, Kaliningrad Branch of Institute of Informatics Problems, Russian Academy of Sciences

**Klemenkov Pavel A.** (b. 1986) — PhD student, Department of System Programming, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

**Kolesnikov Alexander V.** (b. 1948) — Doctor of Science in technology; professor, Immanuel Kant Baltic Federal University; senior scientist, Kaliningrad Branch of Institute of Informatics Problems, Russian Academy of Sciences

**Korenkov Vladimir V.** (b. 1953) — Candidate of Science (PhD) in physics and mathematics; Director, Laboratory of Information Technologies, Joint Institute for Nuclear Research (JINR); Head of Department, International University of Nature, Society and Man "Dubna"

**Korolev Victor Yu.** (b. 1954) — Doctor of Science in physics and mathematics, professor, Department of Mathematical Statistics, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University; leading scientist, Institute of Informatics Problems, Russian Academy of Sciences

**Korolyov Andrey K.** (b. 1992) — student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

**Kovalyov Sergey P.** (b. 1972) — Candidate of Science (PhD) in physics and mathematics, senior scientist, Institute of Control Problems, Russian Academy of Sciences

**Kozhunova Olga S.** (b. 1982) — Candidate of Science (PhD) in technology, Head of Laboratory, Institute of Informatics Problems, Russian Academy of Sciences

**Kozlov Ilya A.** (b. 1989) — MD student, Department of Informatics and Control Systems, Bauman Moscow State Technical University

**Krylov Michael N.** (b. 1992) — student, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

**Kuznetsov Leonid A.** (b. 1942) — Doctor of Science in technology, professor, Head of Department, Russian Presidential Academy of National Economy and Public Administration (Lipetsk Branch)

**Kuznetsova Vera F.** (b. 1948) — Candidate of Science (PhD) in technology, associate professor of the Russian Presidential Academy of National Economy and Public Administration (Lipetsk Branch)

**Listopad Sergey V.** (b. 1984) — Candidate of Science (PhD) in technology, scientist, Kaliningrad Branch of Institute of Informatics Problems, Russian Academy of Sciences

**Mashechkin Igor V.** (b. 1956) — Doctor of Science in physics and mathematics, professor, Facul-

ty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

**Nechaevskiy Andrey V.** (b. 1982) — programmer, Laboratory of Information Technologies, Joint Institute for nuclear research (JINR)

**Petrovskiy Michael I.** (b. 1975) — Candidate of Science (PhD) in physics and mathematics, associate professor, Faculty of Computational Mathematics and Cybernetics, M. V. Lomonosov Moscow State University

**Shkotin Alexander V.** (b. 1952) — software engineer, GIS Department, State Geological Museum of Russian Academy of Sciences

**Simakov Konstantin V.** (b. 1980) — Candidate of Science (PhD) in technology, senior scientist, Bauman Moscow State Technical University

**Stupnikov Sergey A.** (b. 1978) — Candidate of Science (PhD) in technology, senior scientist, Institute of Informatics Problems, Russian Academy of Sciences

**Trofimov Vladimir V.** (b. 1955) — leading programmer, Laboratory of Information Technologies, Joint Institute for nuclear research (JINR)

**Zaks Lily M.** (b. 1989) — principal officer, Department of Modeling and Mathematical Statistics, Alpha-Bank


# XII Всероссийское совещание по проблемам управления

16–19 июня 2014 г.

Институт проблем управления имени В. А. Трапезникова РАН

Москва, Россия

XII Всероссийское совещание по проблемам управления (ВСПУ XII), посвященное 75-летию Института проблем управления (ИПУ) имени В. А. Трапезникова РАН, проводится 16–19 июня 2014 г. в ИПУ РАН (г. Москва, Россия). ВСПУ XII организуется ИПУ РАН при поддержке РФФИ, Отделения энергетики, машиностроения, механики и процессов управления Российской академии наук, Российского национального комитета по автоматическому управлению, Академии навигации и управления движением, Научного совета РАН по комплексным проблемам управления и автоматизации, Совета по мехатронике и робототехнике РАН. Официальный язык Совещания — русский.

**Направления работы**

1. Теория систем управления
2. Управление подвижными объектами и навигация
3. Интеллектуальные системы управления
4. Управление в промышленности, транспортом и логистикой
5. Управление системами междисциплинарной природы
6. Средства измерения, вычислений и контроля в управлении
7. Системный анализ и принятие решений в задачах управления
8. Информационные технологии в управлении
9. Проблемы образования в области управления: современное содержание и технологии обучения

Подробная информация о Совещании находится на сайте http://vspu2014.ipu.ru. Срок окончательной подачи докладов через систему подачи докладов на сайте — **30 ноября** 2013 г.

# Правила подготовки рукописей для публикации в журнале «Информатика и её применения»

1. В журнале печатаются статьи, содержащие результаты, ранее не опубликованные и не предназначенные к одновременной публикации в других изданиях.

   Публикация не должна нарушать закон об авторских правах.

   Направляя рукопись в редакцию, авторы сохраняют все права собственников данной рукописи и при этом передают учредителям и редколлегии неисключительные права на издание статьи на русском языке (или на языке статьи, если он отличен от русского) и на ее распространение в России и за рубежом. Авторы должны представить в редакцию письмо в следующей форме:

   ***Соглашение о передаче права на публикацию:***

   «*Мы, нижеподписавшиеся, авторы рукописи* «. . . », *передаем учредителям и редколлегии журнала «Информатика и её применения» неисключительное право опубликовать данную рукопись статьи на русском языке как в печатной, так и в электронной версиях журнала. Мы подтверждаем, что данная публикация не нарушает авторского права других лиц или организаций*.

   *Подписи авторов: (ф. и. о., дата, адрес)*».

   Это соглашение может быть представлено в бумажном виде или в виде отсканированной копии (с подписями авторов).

   Редколлегия вправе запросить у авторов экспертное заключение о возможности публикации представленной статьи в открытой печати.

2. К статье прилагаются данные автора (авторов) (см. п. 8). При наличии нескольких авторов указывается фамилия автора, ответственного за переписку с редакцией.

3. Редакция журнала осуществляет экспертизу присланных статей в соответствии с принятой в журнале процедурой рецензирования.

   Возвращение рукописи на доработку не означает ее принятия к печати.

   Доработанный вариант с ответом на замечания рецензента необходимо прислать в редакцию.

4. Решение редколлегии о публикации статьи или ее отклонении сообщается авторам. Редколлегия может также направить авторам текст рецензии на их статью. Дискуссия по поводу отклоненных статей не ведется.

5. Редактура статей высылается авторам для просмотра. Замечания к редактуре должны быть присланы авторами в кратчайшие сроки.

6. Рукопись предоставляется в электронном виде в форматах MS WORD (.doc или .docx) или LaTeX (.tex), дополнительно — в формате .pdf, на дискете, лазерном диске или электронной почтой. Предоставление бумажной рукописи необязательно.

7. При подготовке рукописи в MS Word рекомендуется использовать следующие настройки.

   Параметры страницы: формат — A4; ориентация — книжная; поля (см): внутри — 2,5, снаружи — 1,5, сверху — 2, снизу — 2, от края до нижнего колонтитула — 1,3.

   Основной текст: стиль — «Обычный», шрифт — Times New Roman, размер — 14 пунктов, абзацный отступ — 0,5 см, 1,5 интервала, выравнивание — по ширине.

   Рекомендуемый объем рукописи — не свыше 20 страниц указанного формата.

   Сокращения слов, помимо стандартных, не допускаются. Допускается минимальное количество аббревиатур.

   Все страницы рукописи нумеруются.

   Шаблоны примеров оформления, представлены в Интернете:

   http://www.ipiran.ru/journal/template.doc.

8. Статья должна содержать следующую информацию на ***русском и английском языках:***

   – название статьи;

   – Ф.И.О. авторов, на английском можно только имя и фамилию;

   – место работы, с указанием города и страны и электронного адреса каждого автора;

– сведения об авторах, в соответствии с форматом, образцы которого представлены на страницах:
http://www.ipiran.ru/journal/issues/2013_07_01/authors.asp и
http://www.ipiran.ru/journal/issues/2013_07_01_eng/authors.asp;

– аннотация (не менее 100 слов на каждом из языков). Аннотация — это краткое резюме работы, которое может публиковаться отдельно. Она является основным источником информации в информационных системах и базах данных; Английская аннотация должна быть оригинальной, может не быть дословным переводом русского текста и должна быть написана хорошим английским языком.

– ключевые слова, желательно из принятых в мировой научно-технической литературе тематических тезаурусов. Предложения не могут быть ключевыми словами.

9. Литература. По включенным в список литературы работам на русском языке информация в списке представляется как в кириллице, так и с использованием латинской транслитерации, а по работам, написанным латиницей, — на языке оригинала.

   Ссылки на литературу в тексте статьи нумеруются (в квадратных скобках) и располагаются в списке литературы в порядке упоминания.

   В списке литературы не должно быть позиций, на которые нет ссылки в тексте статьи.

10. Присланные в редакцию материалы авторам не возвращаются.

11. При отправке файлов по электронной почте просим придерживаться следующих правил:

   – указывать в поле subject (тема) название журнала и фамилию автора;

   – использовать attach (присоединение);

   – в состав электронной версии статьи должны входить: файл, содержащий текст статьи, и файл(ы), содержащий(е) иллюстрации.

12. Журнал «Информатика и её применения» является некоммерческим изданием. Плата за публикацию не взимается, гонорар авторам не выплачивается.

**Адрес редакции:** Москва 119333, ул. Вавилова, д. 44, корп. 2, ИПИ РАН
Тел.: +7 (499) 135-86-92 Факс: +7 (495) 930-45-05
e-mail: rust@ipiran.ru (Сейфуль-Мулюков Рустем Бадриевич)

# Requirements for manuscripts submitted to Journal "Informatics and Applications"

1. The Journal publishes original articles which have not been published before and are not intended for publication in other editions. An article submitted to the Journal must not violate the Copyright law. Sending the manuscript to the Editorial Board, the authors retain all rights of the owners of the manuscript and transfer the nonexclusive rights to publish the article in Russian (or the language of the article, if not Russian) and its distribution in Russia and abroad to the Founders and the Editorial Board. Authors should submit a letter to the Editorial Board in the following form:

   ***Agreement on the transfer of rights to publish:***

   «*We, the undersigned authors of the manuscript ". . . ," pass to the Founder and the Editorial Board of the Journal "Informatics and Applications" the nonexclusive right to publish the manuscript of the article in Russian (or English) in both print and electronic versions of the Journal. We affirm that this publication does not violate the Copyright of other persons or organizations.*

   *Author(s) signature(s): (name(s), address(es), date)*».

   This agreement should be submitted in paper form or in the form of a scanned copy (signed by the authors).

   The Editorial Board has the right to request from the authors an official expert conclusion that the submitted article does not have secret data prohibited for publication.

2. A submitted article should be attached with **the data on the author(s)** (see p. 8). If there are several authors, the contact person should be indicated who is responsible for correspondence with the Editorial Board.

3. The Editorial Board of the Journal examines the article according to the established reviewing procedure. If authors receive their article for correction after reviewing it does not mean that the article is approved to be published. The corrected article should be sent to the Editorial Board for the subsequent review and approval.

4. The decision on the article publication or its rejection is communicated to the authors. The Editorial Board may also send the reviews on the submitted articles to the authors. Any discussion upon the rejected articles is not possible.

5. The edited articles will be sent to the authors for proofread. The comments of the authors to the edited text of the article should be sent to the Editorial Board as soon as possible.

6. The manuscript of the article should be presented electronically in the MS WORD (.doc or .docx) or LaTeX (.tex) formats and, additionally, in the .pdf format. All documents may be sent by e-mail or on a CD or diskette. A hard copy submission is not necessary.

7. The recommended typesetting instructions for manuscript.

   Pages parameters: format A4, portrait orientation, document margins (cm): left — 2.5, right — 1.5, above — 2.0, below — 2.0, footer 1.3.

   Text: font — Times New Roman, font size — 14, paragraph — 0.5, line spacing — 1.5, justified alignment.

   The recommended manuscript size: no more than 20 pages of the specified format.

   Word abbreviations are not allowed except the standard ones.

   Abbreviations should be minimal. All pages of the manuscript should be numbered.

   The templates for the manuscript typesetting are presented on site:

   http://www.ipiran.ru/journal/template.doc.

8. Articles should enclose data both in **Russian and English**:

   – title;

   – author(s) name(s) and surname(s);

   – affiliation — organization, its address with ZIP code, city, country, and e-mail address;

   – data on authors according to the format (see site):
   http://www.ipiran.ru/journal/issues/2013_07_01/authors.asp and
   http://www.ipiran.ru/journal/issues/2013_07_01_eng/authors.asp;

   – abstract (not less than 100 words) both in Russian and in English. Abstract is a short summary of the article that can be published separately from the article. The abstract is the main source of information on the article and it could be included in leading information systems and data bases. The abstract in English has to be an original text and should not be an exact translation of the Russian one. Good English is required;

– indexing is performed on the basis of key words. The use of key words from the internationally accepted thematic Thesauri is recommended.

  Important! Key words must not be sentences.

9. References. Russian references have to be presented both in Cyrillic and Latin transliteration.

   References in Latin transcript are presented in original language.

   References in the text are numbered according to the order of the appearance and the number is placed in square bracket. References absent in the text should not be included into the list of references.

10. Manuscripts and additional materials are not returned to authors by the Editorial Board.

11. Submissions of files by e-mail must include:

    – the journal title and author(s) name(s) in the "Subject" field;

    – an article and additional materials have to be attached using the "attach" function;

    – an electronic version of the article should contain the file with the text and (a) separate file(s) with figures.

12. "Informatics and Applications" Journal is not a profit publication. There are no charges for the authors as well as there are no royalties.

**Editorial Board address:** 119333, IPIRAN, Vavilova St., 44, block 2, Moscow, Russia
Ph.: +7(499) 135 8692, Fax: +7 (495) 930 4505
e-mail: rust@ipiran.ru (To Prof. Rustem Seyfoul-Mulyukov)

Технический редактор Л. Кокушкина
Художественный редактор М. Седакова
Сдано в набор 01.07.13. Подписано в печать 19.09.13. Формат 60 x 84 / 8
Бумага офсетная. Печать цифровая. Усл.-печ. л. 17,25. Уч.-изд. л. 15,0. Тираж 100 экз.

Заказ № 3951

Издательство «ТОРУС ПРЕСС», Москва 121614, ул. Крылатская, 29-1-43
torus@torus-press.ru; http://www.torus-press.ru

Отпечатано в Академиздатцентре «Наука» РАН с готовых файлов
Москва 121099, Шубинский пер., д. 6