

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

12(184)
2011

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

УЧРЕДИТЕЛЬ
Издательство "Новые технологии"

СОДЕРЖАНИЕ

WEB-ТЕХНОЛОГИИ

- Васенин В. А., Шундеев А. С. Развитие веб-технологий и промежуточного программного обеспечения 2
Тарнавский Г. А., Чесноков С. С. Анализ клиентской базы Центра компьютерного моделирования, "облачного" портала Интернета 9

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Рогозов Ю. И., Свиридов А. С., Дегтярев А. А. Анализ и перспективы развития разработки программного обеспечения 16
Лидовский В. В. Анализ LR-разбора для 18 языков программирования. 21

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Дербенев Н. В., Толчеев В. О. Выявление нечетких дубликатов в наукометрическом анализе 24
Юсупова Н. И., Валеев Р. С. Рациональное размещение грузов в контейнеры с учетом их физических характеристик с помощью роботизированного комплекса . . . 30
Болховитянов А. В., Гусев С. В., Чеповский А. М. Модель и база знаний глагольного управления в предложениях на русском языке 37

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И ДИАГНОСТИЧЕСКИЕ СИСТЕМЫ

- Кузнецов Л. А. Меры оценки состояния объектов и процессов при мониторинге . . 40
Иванов И. А., Увайсов С. У. Информационная модель процесса проектирования контролепригодных радиоэлектронных средств 45
Сперанский Д. В. Идентификация состояний одного типа билинейных систем . . 48

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

- Фролов С. В., Горбунов А. В., Лядов М. А., Фареев С. Г. Блочный-иерархический подход для построения базы знаний информационной системы мониторинга состояния головного мозга человека 54
Садыков С. С., Белякова А. С. Математические модели некоторых сердечно-сосудистых заболеваний 59

ИСТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

- Ковач Г., Шилов В. В. М-3: к истории компьютера первого поколения 64
Указатель статей, опубликованных в журнале "Информационные технологии" в 2011 г. 73
Указатель приложений к журналу "Информационные технологии", опубликованных в 2011 г. 78
Contents 79
Приложение. Сигарев А. А. Концепция построения MPP систем с линейным ростом ускорений.

Главный редактор
НОРЕНКОВ И. П.

Зам. гл. редактора
ФИЛИМОНОВ Н. Б.

Редакционная
коллегия:

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

Редакция:

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

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования.
Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

УДК 004.75 + 004.4

В. А. Васенин, д-р физ.-мат. наук, проф.,
МГУ им. М. В. Ломоносова,
e-mail: vassenin@msu.ru,

А. С. Шундеев, канд. физ.-мат. наук, вед. науч. сотр.,
НИИ механики МГУ им. М. В. Ломоносова,
e-mail: shundeev@msu.ru

Развитие веб-технологий и промежуточного программного обеспечения

Настоящая статья посвящена обзору веб-технологий, анализу их влияния на создание промежуточного программного обеспечения, а также прогнозу их дальнейшего развития.

Ключевые слова: промежуточное программное обеспечение, веб-сервис, стиль REST

Введение

Инфраструктура World Wide Web и составляющие ее основу веб-технологии оказывают большое влияние на все сферы общественных отношений, включая материальную, социальную, политическую, духовную и, конечно, информационную. Есть все основания полагать, что в будущем оно только усилится. В связи с этим представляет интерес ретроспективный анализ развития веб-технологий в целях оценки тенденций их эволюции на ближайшее будущее.

Технологии, реализуемые в виде *промежуточного программного обеспечения (middleware)* [1], занимают центральное место в области построения распределенных программных систем [2]. По определению каждая такая система состоит из набора компонентов, которые выполняются с использованием отдельных процессов операционной системы. Такие процессы могут выполняться как на одном, так и на разных компьютерах. В результате при построении распределенной программной системы приходится решать две, вообще говоря, различные задачи. Первая задача — это реализация межпроцессного взаимодействия, а вторая — реализация бизнес-логики приложения. Использование промежуточного программного обеспечения направлено на решение первой задачи.

Исторически первым примером промежуточного программного обеспечения можно считать *программ-*

ные гнезда (sockets), реализованные в 1982 г. в операционной системе UNIX BSD 4.1 [3]. Данный механизм скрывал неоднородности, которые могли возникать при непосредственном использовании сетевых протоколов и взаимодействии со средой передачи данных. К концу прошлого столетия был предложен целый набор усовершенствованных решений, связанных с построением и использованием промежуточного программного обеспечения [1]. Однако существенное влияние на данную область оказали веб-технологии.

В развитии веб-технологий условно можно выделить три основных этапа. Первый этап (1992—2000 гг.) относится к появлению и становлению глобальной веб-инфраструктуры и формированию ее современного вида. Завершение этого этапа ознаменовалось разработкой базовых средств (фундамента) данной технологии, а также ее теоретического обоснования. Технологическим фундаментом стали протокол HTTP (HyperText Transfer Protocol) [4] и спецификация URI (Uniform Resource Identifier) [5]. Теоретическим обоснованием стал архитектурный стиль REST (REpresentational State Transfer), предложенный Роем Филдингом в диссертационной работе [6].

С момента своего возникновения веб-технологии стали оказывать заметное влияние на индустрию разработки и использования промежуточного программного обеспечения. Так, уже в 1998 г. появилась технология XML-RPC [7], реализующая на базе языка XML (eXtensible Markup Language) [8] и протокола HTTP механизм вызова удаленных процедур. К этому периоду можно отнести и возникновение понятия *веб-сервиса*. Под веб-сервисом понимался серверный компонент распределенной программной системы, обладающий интерфейсом с фиксированными функциями и использующий протокол HTTP для сетевого взаимодействия с клиентами.

Период с 2000 г. по 2006 г., связанный с бурным развитием больших веб-сервисов¹, принято считать вторым значимым этапом развития веб-технологий. В качестве основной цели, стоявшей при создании технологии больших веб-сервисов, рассматривалось построение унифицированной платформы для интеграции приложений, в том числе — унаследованных.

¹ Под большими веб-сервисами понимается их реализация на основе набора спецификаций XML [8], SOAP [9], WSDL [10], BPEL [11].

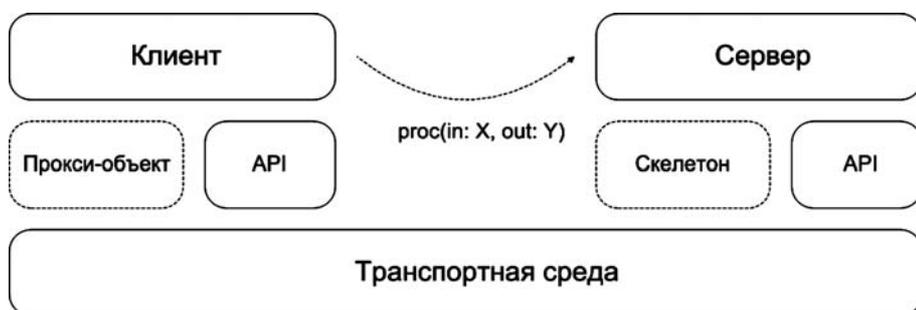
С 2006 г. начался новый этап в развитии веб-технологий, связанный с построением распределенных программных систем на базе RESTful веб-сервисов [7]. Появление данного подхода было обусловлено потребностями, возникшими в индустрии электронной коммерции. При этом в качестве основной решалась задача модернизации с минимальными затратами существовавших на то время веб-сайтов электронной коммерции в веб-сервисы. Целью было расширение сети потребителей за счет привлечения посредников, перепродающих товары и услуги, представленные на основном сайте.

На настоящее время разработанные и прошедшие практическую апробацию решения, относящиеся к трем перечисленным этапам развития веб-технологий, не потеряли своей актуальности и иногда используются совместно. Настоящая статья посвящена обзору веб-технологий, анализу их влияния на построение промежуточного программного обеспечения, а также прогнозу их дальнейшего развития.

Удаленные вызовы процедур

При построении распределенных программных систем наиболее распространенным является следующий сценарий межпроцессного взаимодействия. Выделяются два процесса, один из которых называется *клиентом*, а другой — *сервером*. Сервер может принимать от клиента данные, обрабатывать эти данные и возвращать результат обработки назад клиенту. На время обработки и до получения результата клиент блокируется. Создается иллюзия того, что клиент удаленно на сервере вызывает выполнение некоторой процедуры. Описанный сценарий межпроцессного взаимодействия был положен в основу промежуточного программного обеспечения, получившего название *вызовы удаленных процедур (Remote Procedure Calls, RPC)*. Впервые механизм RPC был реализован в 1984 г. компанией Sun Microsystems [3].

Кратко рассмотрим общую архитектуру механизма RPC, схематично представленную на рисунке. Данную схему можно назвать "канонической" для RPC. Она реализована в таких технологиях, как CORBA (Common Object Request Broker Architecture) [12] и ICE (Internet Communications Engine) [13].



Архитектура удаленного вызова процедур

Для описания интерфейса сервера, как правило, используется специализированный язык *описания интерфейсов (Interface Definition Language, IDL)*. Подобный язык должен предоставлять возможность описания типов данных, используемых в качестве аргументов вызова и результатов выполнения удаленных процедур, а также — сигнатур этих процедур.

В табл. 1 приведен пример подобного описания, в котором в качестве IDL выбран язык Slice (Specification Language for ICE), используемый в программном средстве ICE. В данном описании интерфейс сервера состоит всего из одной процедуры ping, получающей на вход два строковых аргумента. Результатом выполнения процедуры является структура данных pingResult, имеющая два поля.

Таблица 1

Пример описания интерфейса на языке Slice

Описание данных	Описание процедур
<pre>struct pingResult { bool flerror; string message; }</pre>	<pre>interface Server { pingResult ping(string weblogname, string weblogurl); }</pre>

В состав инструментария промежуточного программного обеспечения класса RPC может входить специализированный компилятор, который обеспечивает трансляцию IDL-описания в тексты программ на некотором языке программирования общего назначения (например, C, C#, Python и др.). Данные тексты программ используются для реализации клиента и сервера. Часть сгенерированного программного кода, используемого для реализации клиента, принято называть *прокси-объектом (proxy)*. Вызов соответствующего метода у прокси-объекта инициирует выполнение удаленной процедуры на сервере. Часть сгенерированного программного кода, используемого для реализации сервера, принято называть *скелетом (skeleton)*. Если прокси-объект представляет собой готовый программный код, не требующий дополнительных модификаций, то скелетон является лишь заготовкой. В случае объектного языка программирования в качестве скелетона может выступать класс, содержащий виртуальные методы, которые соответствуют удаленным процедурам. Для реализации сервера подобные виртуальные методы должны быть переопределены.

Существенный недостаток ранних механизмов RPC заключался в том, что они позволяли

создавать только однородные распределенные системы, компоненты которых реализовывались на одном языке программирования (как правило — С) и могли выполняться только под одной операционной системой (как правило — семейства Unix). В результате к середине 1990-х годов назрела острая необходимость в создании промежуточного программного обеспечения типа RPC, позволяющего создавать гетерогенные распределенные системы, охватывающие наиболее распространенные языки программирования и операционные системы. Результатом стало создание технологии CORBA, работы над которой велись в рамках консорциума OMG (Object Management Group). В 1997 г. была опубликована спецификация CORBA 2.0, в которой поддерживалось отображение IDL в языки программирования С и С++, а в 1998 г. было добавлено отображение в язык Java.

Несмотря на большие ожидания, которые возлагались в свое время на CORBA, данный проект нельзя назвать удачным. Надежды, которые возлагались на данную технологию, не были оправданы [12]. Одна из причин этой неудачи была чисто организационной. Различными комитетами и рабочими группами консорциума OMG вносились и принимались в некоторых случаях противоречащие друг другу архитектурные и технологические решения. При этом для принятия вносимого предложения не требовалось предъявления работающего эталонного прототипа или макета, подтверждающего работоспособность предлагаемого решения. Отметим, что впоследствии ряд участников рабочих групп консорциума OMG основали компанию

ZeroC. В рамках этой компании уже на новом технологическом уровне была успешно реализована технология ICE, которую можно считать преемницей CORBA. Технология ICE заняла свою нишу. Ее распространению, в частности, способствовало то обстоятельство, что она базируется на программном обеспечении с открытым исходным кодом. Вместе с тем, широкого признания, которое предсказывали технологии CORBA, ее преемница так и не получила.

Другую причину неудачи CORBA можно объяснить тем, что крупные производители программного обеспечения настороженно и недоверчиво отнеслись к этой технологии. Отметим, что в то же самое время компания Microsoft продвигала свою альтернативу — технологию DCOM (Distributed Component Object Model). Эта технология, так же как и CORBA, являлась промежуточным программным обеспечением класса RPC и реализовывала схожую идею построения распределенных объектов. Если изначально DCOM ограничивался средой Windows, то впоследствии осуществлялись попытки ее портирования под операционные системы семейства Unix. Однако эта платформа так и не стала успешной альтернативой CORBA, и работы над ней были прекращены.

Технология XML-RPC

Как уже было отмечено ранее, в 1998 г. (почти одновременно с технологией CORBA), появилась технология XML-RPC. В отличие от CORBA технология XML-RPC удивительно проста, что обеспечило ее использование на практике и до настоя-

Таблица 2

Пример сообщений XML-RPC

Входное сообщение	Выходное сообщение
<pre>POST /RPC2 HTTP/1.0 Host: rpc.weblogs.com Content-Type: text/xml Content-length: 250 <?xml version="1.0"?> <methodCall> <methodName>ping</methodName> <params> <param> <value>Scripting News</value> </param> <param> <value> http://www.scripting.com/ </value> </param> </params> </methodCall></pre>	<pre>HTTP/1.1 200 OK Content-length: 333 Content-Type: text/xml Date: Sun, 30 Sep 2001 20:02:30 GMT <?xml version="1.0"?> <methodResponse> <params> <param> <value> <struct> <member> <name>flerror</name> <value> <boolean>0</boolean> </value> </member> <member> <name>message</name> <value>Thanks for the ping.</value> </member> </struct> </value> </param> </params> </methodResponse></pre>

шего времени. Основными характеристиками XML-RPC являются следующие.

В качестве транспортного механизма используется протокол HTTP. Методом POST клиент передает входное сообщение на сервер и в синхронном режиме получает ответное сообщение. Для разметки сообщений используется язык XML. Входные сообщения в интуитивно понятном виде задают имя удаленной процедуры и набор входных параметров, а выходные сообщения — результат выполнения удаленной процедуры. Кроме примитивных типов данных поддерживаются неименованные структуры и списки. В табл. 2 представлен пример входных и выходных сообщений XML-RPC.

Технология XML-RPC предлагает новую схему использования механизма RPC, отличную от ранее рассмотренной канонической. Язык XML, выступающий в качестве внешнего (аппаратно и программно независимого) формата данных, благодаря его хорошему восприятию на интуитивном уровне, делает излишними многие традиционные атрибуты RPC. Появилась возможность отказаться от встроенных модулей, реализующих преобразование данных из внутренних форматов во внешний формат и обратно, а также — от модулей осуществляющих автоматическую генерацию прокси-объектов. Излишним стало использование языка описания интерфейсов, который может быть с успехом заменен набором схем XML-данных, описывающих входные и выходные сообщения.

Большие веб-сервисы

Технологии CORBA и XML-RPC являются предшественниками больших веб-сервисов — технологии, ставшей на долги годы лидером в области промежуточного программного обеспечения. Период с 2000 г. по 2006 г., связанный с бурным развитием

больших веб-сервисов, принято считать вторым значимым периодом развития веб-технологий.

Под большими веб-сервисами понимается реализация взаимосвязанного набора спецификаций, базовыми из которых являются SOAP (Simple Object Access Protocol) [9], WSDL (Web Services Description Language) [10], BPEL (Business Process Execution Language) [1], UDDI (Universal Description Discovery and Integration) [14].

Спецификация SOAP определяет стандартный XML-формат для представления сообщений, которыми могут обмениваться компоненты распределенной программной системы. При этом могут реализовываться два архитектурных шаблона построения распределенной системы, а именно — удаленные вызовы процедур, рассмотренные ранее, а также *сервисная шина предприятия (enterprise service bus)*. В качестве транспортного механизма для передачи сообщений обычно используются протоколы HTTP или SMTP (Simple Mail Transfer Protocol) [15].

В табл. 3 представлены примеры SOAP-сообщений, реализующих вызов удаленной процедуры. В качестве транспорта используется протокол HTTP.

Имеет смысл сравнить представленные сообщения с сообщениями из табл. 2. Сообщения протокола SOAP являются более сложными, чем аналогичные сообщения протокола XML-RPC. Следует отметить, что с точки зрения представления различных структур данных выразительные возможности SOAP превосходят XML-RPC. В принципе, любую структуру данных, которая может быть определена на современном языке программирования общего назначения, можно закодировать в SOAP-сообщении. Этот факт свидетельствует о том, что изначально разработчиками предполагался автоматический режим формирования и разбора SOAP-со-

Таблица 3

Пример сообщений SOAP

Входное сообщение	Выходное сообщение
<pre>POST /weblogUpdates HTTP/1.0 Host: rpc.weblogs.com Content-Type: text/xml SOAPAction: "/weblogUpdates" <?xml version="1.0"?> <SOAP-ENV:Envelope ...> <SOAP-ENV:Body> <ping> <weblogname xsi:type="xsd:string"> Scripting News </weblogname> <weblogurl xsi:type="xsd:string"> http://www.scripting.com/ </weblogurl> </ping> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>	<pre>HTTP/1.1 200 OK Content-Type: text/xml Date: Mon, 01 Oct 2001 14:09:02 GMT <?xml version="1.0"?> <SOAP-ENV:Envelope ...> <SOAP-ENV:Body> <pingReasponce> <result> <flerror xsi:type="xsd:boolean"> 0 </flerror> <message xsi:type="xsd:string"> Thanks for the ping. </message> </result> </pingReasponce> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>

Пример описания интерфейса на языке WSDL

Описание данных	Описание процедур
<pre> <types> <s:complexType name="pingResult"> <s:sequence> <s:element minOccurs="1" maxOccurs="1" name="flerror" type="s:boolean"/> <s:element minOccurs="1" maxOccurs="1" name="message" type="s:string"/> </s:sequence> </s:complexType> </types> </pre>	<pre> <message name="pingRequest"> <part name="weblogname" type=s:string"/> <part name="weblogurl" type=s:string"/> </message> <message name="pingResponce"> <part name="result" type=pingResult"/> </message> <portType name="pingPort"> <operation name="ping"> <input message="pingRequest"/> <output message="pingResponce"/> </operation> </portType> </pre>

общений. Для этой цели был разработан язык описания веб-сервисов WSDL.

Описание на языке WSDL можно разделить на две логические части. Первая часть представляет собой описание интерфейса, а вторая — описание самого сервиса. Изначально в спецификации WSDL для обозначения интерфейса использовался термин *тип порта (port type)*. Однако впоследствии, в версии WSDL 2.0 [16] стал употребляться более привычный термин "интерфейс", который и будет использоваться в этой работе. В табл. 4 представлен пример описания интерфейса, функционально эквивалентный описанию из табл. 1, выполненному на языке Slice.

Описание интерфейса включает в себя набор определений операций, поддерживаемых интерфейсом, в том числе — имя каждой операции, определение входных и выходных сообщений, а также структур данных, на базе которых и формируются подобные сообщения. Каждый сервис может одновременно поддерживать несколько интерфейсов. При определении сервиса каждый из его интерфейсов привязывается к конкретному транспортному протоколу, а также к сетевому адресу, по которому клиент может взаимодействовать с соответствующим интерфейсом сервиса.

Даже на основе представленного выше краткого фрагмента WSDL-описания можно сделать вывод, что данный язык является сложным для непосредственного использования (ручного проектирования) без привлечения сторонних средств автоматизации разработки. Одна из распространенных практик использования WSDL сводится к следующему. Инструментальные программные средства позволяют разработчику реализовать методы интерфейса веб-сервиса на каком-то языке программирования общего назначения. Эти же средства позволяют в автоматическом режиме сформировать WSDL-описание, которое может быть опубликовано или передано внешним заинтересованным лицам.

Одна из важных задач, для решения которой разрабатывалась технология больших веб-сервисов, заключается в интеграции унаследованных программных систем, которые используются в организации и были созданы в разное время, на основе разных архитектурных и технологических принципов. Фактически веб-сервис представляет собой обертку (wrapper), представляющую стандартизованный интерфейс (с точки зрения транспортного протокола и форматов представления данных) для взаимодействия с унаследованной системой. Таким образом, задача описания и автоматизации схем взаимодействия между интегрируемыми системами может быть сведена к аналогичной задаче для веб-сервисов. С этой целью был разработан язык WPEL, позволяющий описывать схемы взаимодействия (бизнес-процессы) веб-сервисов и оформлять подобные описания в виде композитных² веб-сервисов.

Кратко рассмотрим спецификацию UDDI. Данная спецификация определяет механизм публикации описаний веб-сервисов в целях их последующего распространения и поиска всеми заинтересованными сторонами (потенциальными потребителями). Данная спецификация является еще более сложной, чем язык WSDL. В 2006 г. компании Microsoft и IBM закрыли свои публичные UDDI-сервисы. Это событие можно рассматривать как знаковое. Назрела потребность пересмотра и, возможно, частичного ухода от чрезмерно сложных, а в некоторых случаях и просто "надуманных" решений, которые были привнесены в индустрию промежуточного программного обеспечения в связи с появлением больших веб-сервисов.

² Композитным веб-сервисом здесь и далее называется сервис, сценарий реализации которого предполагает автоматизированное исполнение других сервисов.

RESTful веб-сервисы

С момента возникновения глобальной веб-инфраструктуры ее важной составляющей стала электронная коммерция. Например, уже в 1995 г. компания Amazon запустила в эксплуатацию свой электронный магазин. Первоначально бизнес-модель электронной коммерции была следующей. Имелся один *поставщик* (продавец) товаров или услуг, по совместительству — владелец электронного магазина. В качестве *потребителей* выступали пользователи, заходившие через веб-браузер на веб-сайт электронного магазина. Таким образом, *сеть распространения* товаров и услуг строилась из числа посетителей одного веб-сайта и ими же ограничивалась.

Потребность в расширении сети распространения вызвала необходимость добавления нового субъекта в существовавшую бизнес-модель электронной коммерции. Подобным субъектом стал *посредник*. Посредник владеет собственным сайтом, через который осуществляет перепродажу исходных услуг и товаров. Возникла необходимость в интеграции веб-сайтов исходного поставщика и посредника. Фактически появилась необходимость в модернизации веб-сайта в веб-сервис с выполнением при этом минимального набора изменений. Создавать параллельно с веб-сайтом новую информационную систему для взаимодействия с посредником представлялось крайне нецелесообразным. В качестве идеальной рассматривалась ситуация, когда изменялся только формат сообщений, которые передаются по протоколу HTTP (например, замена HTML на XML). Теоретически, итоговые веб-сервисы должны были удовлетворять архитектурному стилю REST. Веб-сервисы, для которых это утверждение справедливо, получили название RESTful веб-сервисов [7].

В настоящее время идея использования целевого приложения одновременно и как веб-сайта, и как веб-сервиса оформилась в стандартное требование к современным программным платформам построения веб-приложений. В этой связи следует упомянуть работу [17], в которой авторы реализуют новаторскую идею построения подобной платформы на базе языка манипулирования XML-данными XQuery [18]. В этой работе обозначенное требование формулируется в явном виде и архитектурно реализуется.

С 2006 г. начался новый период в развитии веб-технологий, связанный с построением распределенных программных систем на базе RESTful веб-сервисов. Опишем основные идеи данного подхода, который можно охарактеризовать как ресурсно-ориентированный.

Программная система представляется в виде набора *ресурсов* — поименованных данных. Взаимодействие с программной системой сводится к манипулированию подобными ресурсами через *унифици-*

рованный интерфейс. Унифицированный интерфейс содержит ограниченный набор типов операции. К числу основных операций относятся — создать/изменить/удалить ресурс, получить состояние ресурса, а также получить список всех ресурсов заданного типа.

В рамках RESTful веб-сервиса для наименования ресурсов используется спецификация URI, а для реализации унифицированного интерфейса — протокол HTTP. В качестве примера рассмотрим гипотетический веб-сервис, позволяющий управлять заданиями, выполняющимися на некоторой высокопроизводительной вычислительной установке. Предположим, что данный сервис располагается по адресу <http://computing.ru>.

Адрес <http://computing.ru/job> может использоваться для именованного ресурса, содержащего список всех выполняющихся (и выполненных) вычислительных заданий. Метод GET используется для получения текущего состояния ресурса, в данном случае — списка заданий. Метод POST используется для создания новых ресурсов. Например, для создания нового вычислительного задания необходимо выполнить метод POST по адресу <http://computing.ru/job>. В результате будет создано новое задание и одновременно ресурс, позволяющий с ним работать. Для именованного этого ресурса естественно использовать адрес вида http://computing.ru/job/{job_id}, где *job_id* — уникальный идентификатор, присваиваемый заданию при его создании. Выполняя по этому адресу метод GET, будет получена информация о текущем состоянии выполнения задания. Метод PUT позволит изменить это состояние (например, приостановить его выполнение). Для уничтожения ресурса (аннулирования задания) необходимо воспользоваться методом DELETE.

Кроме базовых методов GET, POST, PUT, DELETE допускается использование HTTP методов HEAD и OPTIONS. Метод HEAD используется для получения метаинформации о ресурсе, а OPTIONS — возвращает список допустимых операций. Например, для ресурса <http://computing.ru/job> допустимыми являются методы GET, POST, в то время как для ресурса http://computing.ru/job/{job_id} допустимыми являются GET, PUT и DELETE.

Операции GET, HEAD должны быть *безопасными*. Их выполнение не должно повлечь за собой изменение состояния ресурса. Операции PUT, DELETE должны быть *идемпотентными*. Их повторное выполнение "игнорируется".

Следует отметить, утверждение, что метод GET возвращает состояние ресурса, не совсем верно. Метод GET возвращает *представление (representation)* состояния ресурса. У одного и того же ресурса может одновременно поддерживаться несколько форматов представлений. Распространенными форматами являются XML, XHTML (eXtensible Hypertext Markup Language) [19], JSON (JavaScript Object

Notation) [20]. Установка предпочтительного формата, а также используемого по умолчанию осуществляется с помощью параметра Content-Type в заголовке HTTP-запроса.

Рекомендуется при создании RESTful веб-сервисов реализовывать связность ресурсов, сохраняя в их представлениях ссылки на другие ресурсы. Например, естественным выглядит наличие ссылки на список заданий в представлении индивидуального задания.

Гибридные веб-сервисы

Учитывая широкую распространенность и признание протокола HTTP, не удивляет тот факт, что до настоящего времени было создано огромное число веб-приложений. Большинство из этих веб-приложений не удовлетворяет архитектурному стилю REST. В некоторых случаях даже веб-сайты архитектурно имеют больше общих черт с технологией XML-RPC, чем со стилем REST. Подобные веб-приложения естественно именовать гибридными веб-сервисами, одновременно содержащими в себе черты RESTful веб-сервисов и использующими технологии XML-RPC. В этом обобщенном разделении большие веб-сервисы естественно рассматривать как разновидность XML-RPC веб-сервисов.

Проанализируем причины появления подобных гибридов. Представленная выше "каноническая" схема построения RESTful веб-сервиса не всегда может быть реализована в силу объективных технических причин. Методы протокола HTTP, отличные от GET и POST, могут просто не поддерживаться. Это произойдет при использовании XHTML-представлений ресурсов. У ряда распространенных веб-серверов имеются ограничения на размер строки URI при выполнении GET-запросов. По этой причине могут возникнуть ситуации, в которых придется обходиться только методом POST.

Изначальный выбор между построением приложения в виде большого веб-сервиса или RESTful веб-сервиса является нетривиальной задачей, о чем свидетельствуют результаты исследований, представленные в работе [21]. В данной работе было показано, что эти два подхода различаются исходными задачами, для решения которых они были разработаны. Каждый из этих подходов имеет право на существование. Нередко разработчики намеренно реализуют для своей программной системы два интерфейса. Этим путем, в частности, пошла компания Amazon [22]. Сервисы данных этой компании поддерживают интерфейс RESTful веб-сервиса, использующий методы SET, PUT и DELETE протокола HTTP, и, одновременно, интерфейс большого веб-сервиса, использующий только метод POST. Таким образом, в данном случае имеет место гибриды.

По всей вероятности, в области веб-технологий будущее как раз за гибридами. Косвенным под-

тверждением этого факта может служить версия WSDL 2.0 [16]. Данная спецификация в явном виде позволяет привязывать интерфейс веб-сервиса к протоколу HTTP, а не только к SOAP.

Заключение

В представленной работе были проанализированы основные этапы развития веб-технологий и их влияние на создание промежуточного программного обеспечения, используемого при построении распределенных программных систем. Полученные результаты будут использованы в следующей работе авторов, посвященной эволюции Грид-технологий.

Необходимо отметить, что основные концептуальные работы по технологии Грид, в первую очередь касающиеся создания Открытой архитектуры Грид-сервисов, относятся к 2001—2006 гг. Как было показано, этот период можно охарактеризовать как "эпоху" больших веб-сервисов, которая ознаменовалась рождением, бурным ростом, а затем и определенной стагнацией этой технологии. Учитывая, что большие веб-сервисы являются технологическим фундаментом для Грид-систем, особую актуальность обретают следующие вопросы. Насколько технология больших веб-сервисов подходит для создания и развития Грид-систем? Имеются ли адекватные альтернативы для ее замены, насколько перспективны альтернативные решения?

Список литературы

1. **Emmerich W., Aoyama M., Sventek J.** The impact of research on middleware technology // ACM SIGSOFT Software Engineering Notes. 2007. V. 32. N 1. P. 89—112.
2. **Таненбаум Э., ван Стеен М.** Распределенные системы. Принципы и парадигмы. СПб.: Питер, 2003. 877 с.
3. **Кузнецов С. Д.** Операционная система UNIX // Центр информационных технологий [Электронный ресурс]. Центр информационных технологий. URL: http://citforum.ru/operating_systems/unix/contents.shtml.
4. **Спецификация HTTP.** Hypertext Transfer Protocol — HTTP/1.1. 1999. URL: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.
5. **Спецификация URL Uniform Resource Identifier (URI): Generic Syntax** 2005. URL: <http://tools.ietf.org/html/rfc3986>, свободный.
6. **Fielding R. T.** Architectural Styles and the Design of Network-based Software Architectures // Dissertation doctor of philosophy in Information and Computer Science. University of California. 2000.
7. **Richardson L., Ruby S.** RESTful Web Services. O'Reilly, 2007. 448 p.
8. **Спецификация XML.** Extensible Markup Language (XML) 1.0 / Fourth Edition. 2006. URL: <http://www.w3.org/TR/2006/REC-xml-20060816/>.
9. **Спецификация SOAP 1.1.** Simple Object Access Protocol (SOAP) 1.1. 2000. URL: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>.
10. **Спецификация WSDL 1.1.** Web Services Description Language (WSDL) 1.1. 2001. URL: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>.
11. **Спецификация BPEL 2.0.** Web Services Business Process Execution Language Version 2.0. 2007. URL: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
12. **Henning M.** The Rise and Fall of CORBA // ACM Queue. 2006. V. 4. N 5. P. 28—34.

13. Henning M., Spruiell M. Distributed Programming with Ice, Revision 3.4 // Distributed Computing with Ice. ZeroC Inc. 2010.
14. Спецификация UDDI. UDDI Version 3.0.2. 2004. URL: <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>.
15. Спецификация SMTP. Simple Mail Transfer Protocol / Network Working Group. 2008. URL: <http://tools.ietf.org/html/rfc5321>.
16. Спецификация WSDL 2.0. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. 2007. URL: <http://www.w3.org/TR/2007/REC-wsdl20-20070626>.
17. Kaufmann M., Kossmann D. Developing an Enterprise Web Application in XQuery // ICWL 2009. LNCS 2009. V. 5648. P. 465—468.
18. Кац Г. и др. W3C XML: XQuery от экспертов. Руководство по языку запросов. М.: Кудиц-образ, 2005. 480 с.

19. Спецификация XHTML 1.1. XHTML 1.1 — Module-based XHTML-Second Edition. 2010. URL: <http://www.w3.org/TR/2010/REC-xhtml1-20101123>, свободный.
20. Спецификация JSON. An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET. 2007. URL: <http://msdn.microsoft.com/en-us/library/bb299886.aspx>, свободный.
21. Pautasso C., Zimmermann O., Leymann F. RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision // Proc. of the 17th International World Wide Web Conference. 2008. P. 805—814.
22. Murty J. Programming Amazon Web Services: S3, EC2, SQS, FPS and SimpleDB. O'Reilly Media. 1 edition. 2008. 608 p.

УДК 004.738.5:001.891.57

Г. А. Тарнавский,

д-р физ.-мат. наук, вед. науч. сотр.,
Институт вычислительной математики
и математической геофизики СО РАН,
г. Новосибирск,

e-mail: Gennady.tarnavsky@gmail.com

С. С. Чесноков, аспирант,

Новосибирский государственный
технический университет,
e-mail: sergey.chesnokov@gmail.com

Анализ клиентской базы Центра компьютерного моделирования, "облачного" портала Интернета

Проанализирован опыт функционирования Центра компьютерного моделирования в Интернете. Рассмотрена статистика посещений Web-портала, пионерного проекта, базирующегося на современных информационных технологиях концепции Cloud Computing ("Облачные вычисления") и предназначенного для коммерческого завершения проведенных программных разработок.

Ключевые слова: Интернет, информационные технологии, компьютерное моделирование, облачные вычисления, статистика посещений

Направление *Cloud Computing* ("Облачные вычисления") — новое явление, призванное изменить отношение к ИТ-индустрии, новая модель, позволяющая предоставлять информационные ресурсы как сервисы (услуги), новая концепция развития отрасли информационных технологий, исключительно перспективная в бизнес-приложениях, науке и инженерии.

Следует подчеркнуть высокую конкурентность развития облачных вычислений (см., например, [1]), поэтому анализ различных аспектов востребованности облачных программных продуктов представляет, на наш взгляд, существенный интерес.

Краткий обзор развития облачной ИТ-индустрии

Первые идеи, косвенно относящиеся к направлению *Cloud Computing* с проработкой возможности вычисления в режиме удаленного доступа, были высказаны в 70—80-х годах прошлого века, однако первые практические реализации этого тренда появились к 2001 г., когда впервые был использован термин "Облачные вычисления" (облако — один из символов Интернета).

С 2006 г. началась интенсификация "облачности", когда корпорация *Amazon* представила свою инфраструктуру Web-сервисов, не только обеспечивающую хостинг, но и предоставляющая клиенту удаленный доступ к своим вычислительным мощностям. Вслед за *Amazon* аналогичные сервисы представили корпорации *Google*, *Sun* и *IBM*. В 2008 г. корпорация *Microsoft* анонсировала не просто сервис, а полноценную операционную систему *Windows Azure*. Ее официальный релиз состоялся в декабре 2010 г., открыв тем самым настоящую войну между операционными системами *Google Chrome*, *Google Android*, *Apple iOS* и *Intel-Nokia MeeGo*¹.

17 июля 2010 г. в США был запущен облачный сервис *OnLive*, предоставляющий возможность играть в современные игры на самом простом оборудовании. Технически это выглядит следующим образом. Сама игра располагается на удаленном сервере и там же проводится обработка графики, которая на компьютер конечному пользователю (игроку) поступает уже в "готовом" виде. Таким образом, те вычисления, которые в обычной игре вы-

¹ Необходимо уточнить, что операционные системы *Google Android*, *Apple iOS* и *MeeGo* следует позиционировать как "полнооблачные", поскольку они в значительной мере развертываются на аппаратной платформе клиента, чем существенно отличаются, например, от *Chrome OS*, которая почти полностью развертывается на серверах *Google*, а в гаджете клиента устанавливается только браузер. Созданный корпорацией *Google* специальный ноутбук *BlackBook* характеризуется минимизацией средств (упрошенная клавиатура с отсутствием некоторых классических клавиш и т. п.), что позволяет расширить рынок продаж с ориентацией на многочисленные "малопродвинутые" круги потребителей.

полняются на компьютере игрока и требуют хороший микропроцессор и мощную видеокарту, здесь уже выполнены на сервере, а компьютер клиента используется лишь как монитор и Интернет-браузер. В ближайшее время станет возможным использовать для самых ресурсо-затратных игр даже простые планшетики и смартфоны.

Помимо игровых развлечений (включая просмотр кинофильмов без их покупки и скачивания и т. п.), облачные вычисления имеют отличные перспективы бизнес-приложений. Эти перспективы и современные реалии были проанализированы на первом международном конгрессе по облачным вычислениям *Cloud Computing Congress CCC-2010* (см. [2]).

Компьютерное моделирование в Интернете

Исключительные перспективы открывает направление *Cloud Computing* для математического моделирования в научных исследованиях и проведения компьютерных расчетов инженерных задач, в особенности связанных с использованием больших программных комплексов (подробнее см. работу [3]).

Традиционно процесс передачи вычислительных комплексов заключался в приобретении потребителем (будущим пользователем) у разработчиков кодов (в некоторых случаях и текстов) программ и инсталляции их у себя на компьютере. При этом для безотказной и декларированной работы программного комплекса пользователь должен иметь аналогичную операционную среду, в том числе системы визуализации цифровых данных. В целом этап инсталляции программ, даже при поддержке разработчика, является весьма сложным и зачастую вызывающим массу затруднений, преодоление которых требует значительных и, в сущности, непродуктивных затрат интеллектуальных усилий и времени. От всех проблем, связанных с инсталляцией приобретенного программного продукта, освобождает новая форма использования вычислительных комплексов. Подчеркнем, что здесь имеется в виду именно использование программных комплексов, а не их продажа. Электронные формы торговли в Интернете, в том числе компьютерными программами, уже достаточно хорошо развиты.

В Интернете создается особая площадка — Центр компьютерного моделирования. В этом Центре размещаются программные комплексы со всеми своими атрибутами — препроцессорной системой подготовки заданий, процессорной системой выполнения заданий (проведения вычислительных операций) и постпроцессорной системой вывода полученной информации в цифровой и графической формах. Пользователь должен только сформулировать собственную вычислительную задачу, введя необходимые числовые данные, и запустить ее на счет. Решение задачи ему будет предоставлено по окончании вычислений. Комфорт пребывания

посетителя в Центре должны обеспечивать специальные сервисы клиентской поддержки. В экономических терминах такой метод эксплуатации программных комплексов означает не покупку, а взятие их в аренду или лизинг у разработчиков ресурса.

Пионером этого нового направления развития современных информационных технологий является Центр компьютерного моделирования *SciShop.ru*, разрабатываемый с 1998 г. и функционирующий в Интернете с 2004 г. (см., например, [4, 5]). Подчеркнем, что Центр компьютерного моделирования — **первый и пока единственный в мире** "облачный" портал Интернета, на котором пользователь (клиент) может самостоятельно провести непосредственное компьютерное моделирование ряда научных и инженерных задач, без предварительной покупки программных комплексов и инсталляции их на собственных вычислительных ресурсах (рабочих станциях, персональных компьютерах и ноутбуках).

Позитивные и негативные аспекты Cloud Computing

На основе проведенной разработки и анализа функционирования Центра компьютерного моделирования определены позитивные и негативные факторы применения облачных технологий для обеспечения научных и инженерных исследований непосредственно в Интернете, без этапа приобретения программных комплексов и инсталляции их на собственных мощностях (центрах коллективного пользования, рабочих станциях, персональных компьютерах).

Позитивные факторы для разработчиков:

- эффективная борьба с нелегальным использованием программного продукта, поскольку сам продукт не попадает к заказчику;
- относительная легкость обнаружения и пресечения несанкционированного использования доступа нескольких пользователей под одним логином;
- существенное уменьшение затрат на развертывание и внедрение технической и консалтинговой поддержки для каждого заказчика.

Позитивные факторы для потребителей:

- отсутствие необходимости установки программного обеспечения на рабочих местах пользователей, поскольку доступ к нему осуществляется через обычный браузер;
- радикальное сокращение затрат на развертывание системы в организации;
- сокращение затрат на техническую поддержку и обновление развернутых систем, вплоть до их полного отсутствия;
- быстрота внедрения, обусловленная отсутствием затрат времени на развертывание системы;
- понятный интерфейс;
- ясность и предсказуемость платежей;

- возможность получения более высокого уровня обслуживания программного обеспечения.

Негативные факторы для разработчиков:

- концепция *Cloud Computing* применима далеко не для всех функциональных задач;
- поскольку основная экономия ресурсов провайдера достигается за счет масштаба, модель облачных вычислений оказывается неэффективной для малого числа клиентов;
- модель неэффективна при необходимости глубокой индивидуальной адаптации под каждого заказчика.

Негативные факторы для заказчиков:

- привязка заказчиков к единственному разработчику и его хостинг-площадке;
- нестабильность работы провайдера может приводить к невозможности долгосрочного планирования и даже срыву сроков проведения работ;
- нежелательность использования облачных вычислений для проектов строгой конфиденциальности вследствие высокой возможности утечки информации со стороны поставщика услуг и невозможность контролировать этот процесс;
- затруднительность повышения качества сервисов в текущем режиме работы;
- необходимость постоянно действующего подключения к Интернету с достаточно высокой скоростью передачи данных.

Приведенные формулировки являются краткими по форме, но глубокими по содержанию и представляют существенную практическую пользу при аргументации *pro* или *contra* в решении вопроса, актуального для каждого разработчика и потребителя, будет ли эффективным использование облачных технологий для продвижения созданных программных продуктов.

Объект и период исследований

Центр компьютерного моделирования в Интернете *SciShop.ru* (далее — Центр) является одним из пионерских проектов по созданию и развитию новых форм передачи завершенных научных разработок — программных комплексов. Главная сущность этих новых форм заключается в том, что потребителю передаются на безвозмездной или коммерческой основе не сами комплексы, а предоставляются сервисы по их эксплуатации. На специализированном Web-портале в режиме дистанционного доступа посетитель имеет возможность решать свои собственные научные задачи — непосредственно самому формировать задание на счет, инициализировать процессорные системы и получать в свое распоряжение численное решение.

Подробнее с организацией обслуживания пользователей в Центре можно ознакомиться в работе [6] или на самом портале в его разделе "Информация о Центре".

Дальнейшее развитие Центра требует постоянного анализа эффективности функционирования, и этот анализ можно проводить по статистике посещений Web-ресурса. В данной статье рассматривается статистика Центра за период 12 месяцев от 1 июля 2008 г. до 30 июня 2009 г.

К анализу будут также привлекаться данные за прошлые учетные периоды — с 1 апреля 2008 г. по 31 марта 2009 г. и с 1 января по 31 декабря 2008 г. Отметим, что учетно-аналитические периоды проводятся каждые 3 месяца с соответствующим 9-месячным перекрытием для того, чтобы статистика носила "преемственный" характер и плавно фиксировала изменения цифровых параметров.

Выбор для анализа именно этого периода связан с тем важным обстоятельством, что статистика более позднего времени (2010—2011), а также динамика изменения ее различных сторон пока носит конфиденциальный характер в связи с высокой конкурентностью процессов расширения облачных технологий для научных и инженерных приложений.

Общие замечания к анализу

Портал *SciShop.ru* является научным, специализированным в области компьютерного моделирования Web-ресурсом. В настоящее время Центр содержит четыре действующих программных комплекса:

- "Удар" — высокоскоростная внутренняя аэродинамика: расчет ударно-волновых структур в диффузоре гиперзвукового прямоточного воздушно-реактивного двигателя [7];
- "Поток" — высокоскоростная внешняя аэродинамика: расчет обтекания объектов в атмосферах Земли и Марса [8];
- "Астра" — вычислительная астрофизика: моделирование динамики процессов в межгалактическом газе и протопланетных облаках [9];
- "Нано" — микроэлектроника: компьютерная поддержка проектирования наноструктурированных полупроводниковых материалов [10].

Посетителю Центра предоставляются следующие сервисы:

- 1) доступ к библиотекам, содержащим научные статьи по декларированным тематикам;
- 2) доступ к табличным и/или графическим базам данных, содержащим результаты компьютерного моделирования большого спектра научных и прикладных задач;
- 3) доступ к процессорным системам, позволяющим клиенту самому организовать и провести компьютерное моделирование интересующих его задач;
- 4) доступ к шлюзам для выхода на сайты ведущих российских и зарубежных журналов в соответствующих областях знания;
- 5) доступ на сегмент "Форум" для получения консультаций и обсуждения проблем.

В некоммерческом режиме посетителю доступны демоверсии систем Центра. Для полномасштабного

доступа следует пройти регистрацию и произвести оплату. Система приема платежей Центра принимает оплату из любых электронных платежных систем (*WebMoney*, Яндекс.Деньги, *E-gold* и др., входящих в консорциум *Robokassa*). Разработана и верифицируется система приема оплаты с использованием кредитных банковских карт и SMS-сообщений сотовой телефонии.

Это обстоятельство (специализацию портала) надо учитывать при анализе числа посещений и времени пребывания клиента в Центре, глубины и ширины их движения по portalу, частоты пользования разделами "Удар", "Поток", "Астра" и "Нано", сегментами каждого из этих разделов — "Библиография", "Процессорные системы" и "Базы данных".

Также следует анализировать степень использования посетителями шлюзов портала для выхода на сайты ведущих международных и отечественных журналов по тематикам "Математическое моделирование", "Физическая газовая динамика" и "Нанотехнологии в электронике". Эффективным средством для улучшения привлекательности сайта и проведения успешного маркетинга является анализ записей в гостевой книге раздела "Форум".

В целом следует подчеркнуть, что, естественно, невозможно ни ожидать "супервысокой" посещаемости портала, ни добиваться любой ценой показателей, сравнимых с показателями посещаемости порталов типа "Одноклассники" или "Facebook".

Анализ тайм-статистики

Важным методом определения эффективности функционирования портала является анализ временного аспекта посещаемости: помесечная статистика за год, посуточная статистика за какой-либо месяц этого года и почасовая статистика за какой-либо день выбранного месяца.

Помесечная статистика посещений сайта с июля 2008 г. по июнь 2009 г. приведена на рис. 1 (см. четвертую сторону обложки).

Анализируя данные, можно высказать следующие соображения. Более высокая активность имеет место в осенние, зимние и весенние месяцы. Число посещений варьируется в пределах от 714 в июле до 1947 в январе, число запросов — от 4081 в июле до 8265 в декабре. Представим некоторые технические характеристики посещений — объем скачанной с сайта информации (текстов статей, результатов расчетов и т. п.). Этот показатель может быть полезен при детальном анализе. В частности, в июле было скачано всего 9 Мбайт информации, а в декабре — 62 Мбайт. Общий объем взятой с сайта за год информации составил около 350 Мбайт.

Число посещений за год — более 17 000, число запросов — около 70 000. За предыдущий период анализа статистики (июль 2007 г.—июнь 2008 г.) эти показатели составили 16 300 и 65 000 соответственно.

Таким образом, можно констатировать стабильное состояние интереса к ресурсу с очень незначительным ростом.

Этот результат был вполне прогнозируем, поскольку активный маркетинг в данный период не проводился вследствие крайне ограниченных финансовых возможностей проекта.

Проанализируем динамику посещений портала в течение одного месяца. На рис. 2 (см. четвертую сторону обложки) показана посуточная статистика за май 2009 г. Май является месяцем "умеренного рабочего типа" (предотпускной период), и график использования портала может дать полезную информацию для повышения уровня сервисов ресурса. Рис. 2 включает в себя три сегмента. Верхний сегмент представляет динамику по дням числа просмотренных страниц и числа сделанных запросов. Средний сегмент иллюстрирует число посещений и число уникальных заходов на портал. Нижний сегмент представляет динамику объема скачанных файлов.

Анализ этих диаграмм приводит к следующим выводам. Число посещений сайта достаточно неравномерно по дням и меняется от 29 посещений (1,7 %, 10 мая, воскресенье) до 79 посещений (4,6 %, 12 мая, вторник) и 82 посещения (4,8 %, 15 мая, пятница). Этот результат, казалось бы, достаточно очевиден — в рабочие дни активность пользователей и должна быть больше. Однако дни посетительской активности 12 и 15 мая не являются ни днями максимального числа запросов, ни днями максимального объема скачанной информации.

Максимум запросов был сделан в пятницу 1 мая (в России — праздничный день), в воскресенье 3 мая и в воскресенье 24 мая.

Объем скачанных файлов достиг максимума 1 мая (2 Мбайта, 11,1 % от всего месячного показателя) и двух менее резких максимумов 3 мая (1,073 Мбайта, 6 %) и 24 мая (1,6 Мбайта, 7,7 %). Таким образом, данные по объему скачанной информации соответствуют данным по числу запросов и не коррелируют с днями максимальной посещаемости сайта.

Этот результат довольно неожиданен и требует размышлений. Возможно, имеет место "праздношатание" по Интернету в выходные дни, но администрация сайта в числе своих посетителей хотела бы видеть "целевую" аудиторию — специалистов в области математического моделирования.

Интересными представляются результаты анализа активности посетителей сайта в различное время суток, т. е. почасовая статистика. Поскольку возможны случайные отклонения в различные дни (рабочие, выходные, праздничные), следует проводить определенное усреднение почасовых данных.

На рис. 3 (см. четвертую сторону обложки) приведена суточная почасовая статистика, усредненная по всем числам мая 2009 г.

Из графиков статистики следует, что по числу сделанных запросов наиболее высокая активность посетителей наблюдалась в период с 6 до 7, с 13 до 14, с 14 до 15, с 16 до 17 и с 18 до 19 часов по московскому времени. В эти периоды число запросов составляло соответственно 385 (6,6 % от всего числа запросов), 386 (6,6 %), 530 (9,1 %), 374 (6,4 %) и 356 (6,1 %) запросов при среднем значении 270 запросов в час.

Этот результат представляется, в общем, естественным — максимум активности соответствует рабочему периоду суток, причем его второй половине, в соответствии со "стандартной" организацией рабочего времени программиста.

Однако следует обратить внимание на второй по интенсивности пик запросов с 6 до 7 часов утра по московскому времени. Его происхождение можно отнести к сдвигу по времени в часовых поясах России: в основных научных восточных центрах этот период соответствует периодам с 9 до 10 часов в Новосибирске, с 11 до 12 часов в Иркутске и т. п.

Статистика числа просмотренных страниц и объем скачанной информации практически полностью соответствует статистике запросов, с максимумом 266 (6,8 %) страниц и 1,5 Мбайта (8,4 %) в период с 14 до 15 часов.

Таким образом, почасовая статистика иллюстрирует максимальную активность на сайте в рабочее время для Центральной России, где сконцентрирован основной научный потенциал страны. Это показывает, что задача, которую ставят перед собой разработчики портала — его посещение именно специалистами, в основных чертах выполнена.

Статистика посещений по странам

Рассмотрим статистические данные о государственной принадлежности посетителей Центра. На рис. 4 (см. четвертую сторону обложки) приведена диаграмма, показывающая статистику посещений "по странам".

Максимальное число посетителей сайта — специалисты из России. Они составляют 66 % от числа всех посетителей, просмотрели 65 % файлов и скачали 69 % от всей востребованной информации.

Этот результат вполне ожидаем для сайта в зоне .ru Интернета. Однако целью разработчиков и администрации ресурса является максимальное "географическое" представление контингента посетителей. Для этого поддерживается (частично, в соответствии с финансовыми возможностями) англоязычное "зеркало" сайта. Разумеется, для достижения данной цели необходима активная маркетинговая политика, но она также требует высоких финансовых вложений.

На втором месте по числу посетителей находятся специалисты из Украины — 8 % от числа всех запросов, 5 % просмотренных файлов и 4 % объема скачанной информации.

Резкое отставание показателей Украины от показателей России косвенно свидетельствует об упадке интереса в этой стране к аэрокосмической и нанотехнологической тематикам, преобладающим в настоящее время в Центре компьютерного моделирования, хотя Украина и имеет высокотехнологичную базу (НИИ, КБ) поддержки производства авиакосмической техники в Днепропетровске и Киеве ("Южмаш" и "Антонов"), на привлечение специалистов из которых частично ориентированы программные комплексы Web-ресурса.

Третье место, как и ожидалось, принадлежит США — 7 % запросов, 9 % просмотренных файлов и 6 % скачанной информации (опережая по файлам и информации Украину). Разумеется, разработчикам сайта хотелось бы видеть более высокие показатели ведущей индустриальной страны мира. В этом направлении имеется большой "непаханный" пласт повышения привлекательности Web-ресурса — ориентация на роботы поисковых машин *Google*, *Yahoo* и т. п., обмен баннерами с научными сайтами, улучшение дизайна англоязычного сегмента.

Четвертое место занимают Нидерланды — 7 % запросов (наравне с США), 8 % просмотренных файлов и 10 % объема скачанной информации. Последний показатель очень высок (превышает аналогичные показатели и Украины, и США) и свидетельствует о неслучайном характере нидерландских визитов на сайт. Этот результат, вследствие отсутствия развитой аэрокосмической промышленности в Нидерландах, является неожиданным и требует размышлений.

Пятое место занимает Германия — 4 % запросов, 6 % просмотренных файлов и 4,5 % скачанной информации. Это ожидаемый результат для такой промышленной страны.

Дальнейшие места занимают "Неопознанные (по государственной принадлежности) входы", Китай, Беларусь, Казахстан и Великобритания.

Отсутствие в первой десятке посещений специалистов из Франции (29 место), одной из ведущих аэрокосмических стран, является неприятным фактом, требует особого анализа и принятия каких-либо мер.

Вторую десятку (11—20 места) составляют Латвия, Италия, Израиль, Молдавия, Бразилия, Канада, Кипр, Греция, Гонконг и Япония.

Отметим, что из стран, развивающих авиакосмическую науку и технику, в этом списке находятся Израиль, Бразилия, Япония и отсутствуют Индия, Пакистан и Иран.

Первые 20 позиций учитываются вследствие того, что, как правило, 20-е место есть некий рубеж, до которого число запросов превышает 1 %. Далее показатель становится слишком малым и анализ может быть некорректным вследствие статистических отклонений. В большинстве случаев — это случай-

ные визиты на сайт, хотя возможны и исключения, одиночные "осознанные" посещения специалистов.

Всего за период с июля 2008 г. по июнь 2009 г. портал посетили специалисты из 43 стран.

В прошлый аналитический период (апрель 2008 г. — март 2009 г.) порталом пользовались специалисты из 36 стран. В тот период первую пятерку стран составляли Россия, "Неопознанные входы", Украина, США и Нидерланды. Отметим, что в этом списке также присутствовали Нидерланды. Таким образом, этот факт не может быть приписан случайным посещениям и статистическим отклонениям.

В первой двадцатке стран в прошлый учетный период присутствовали Германия, Китай, Великобритания, Израиль, Тайвань и отсутствовали "молодые тигры" — Япония, Индия, Пакистан, Иран и Бразилия.

В позапрошлый аналитический период (январь—декабрь 2008 г.) сайт посетили специалисты из 45 стран. Это ровный показатель, изменяющийся в диапазоне отклонений не более 15 %.

Первую пятерку составили почти те же страны: Россия, США, "Неопознанные входы", Украина и Германия. Укажем, что Нидерланды в тот период занимали 11 место.

В первую двадцатку стран также вошли (из интересующих и отслеживаемых стран) Великобритания (10 место), Китай (13), Франция (14), Бразилия (18), Пакистан (19) и не вошли Япония (22) и Израиль (27).

Иран и в тот период не фигурировал в списке. Однако аэрокосмическая промышленность этой страны развивается интенсивными темпами (уже запущен собственный спутник Земли). Поэтому следует предположить, что иранские специалисты, возможно, входят в Интернет таким образом, что попадают в статистику учета под псевдонимом "Неопознанные входы", которые всегда занимают места в первой пятерке стран.

Анализ данных за три учетных периода

Приведем сводные данные за три 12-месячных периода:

- с 1 января по 31 декабря 2008 г.
- с 1 апреля 2008 г. по 31 марта 2009 г.
- с 1 июля 2008 г. по 30 июня 2009 г.

В таблице приведены основные статистические данные следующих параметров за указанные учетные периоды:

- 1) общее число посещений сайта;
- 2) число посещенных страниц;
- 3) число просмотренных файлов;
- 4) число сделанных запросов;
- 5) объем скачанной информации (Мбайт);
- 6) число стран-резидентов посетителей.

Анализ данных, представляющих итоги функционирования ресурса в Интернете и сведенных в таблицу, может дать весьма полезные результаты

Статистика за три периода

Параметр	Период		
	I	II	III
1	14941	16235	17204
2	31633	35056	39023
3	40480	31309	33292
4	73273	64288	67873
5	1316	415	333
6	45	36	43

для повышения эффективности решения задач, которые поставлены в проекте.

Цифры, приведенные в таблице, показывают стабильность уровня посещаемости Центра и работы клиентов в нем. Так, параметры в каждой строке отличаются друг от друга не более, чем на 20 %. Исключением является только объем скачанной информации в период I, существенно превышающий аналогичные показатели за периоды II и III.

В целом статистика дает основания к выводу об успешности первого этапа проекта с хорошими перспективами на будущее.

Так, среднее ежедневное число посещений на протяжении последних 1,5 лет составило около 50 визитов, среднее число просмотренных страниц — около 100, среднее число запросов — более 180 в день.

Поскольку данный Web-ресурс не является ни развлекательным, ни публицистическим порталом, а специализирован в достаточно узкой научной области, то эти результаты являются вполне приемлемыми.

Однако следует кратко затронуть коммерческую сторону проекта. Поскольку большинство ресурсов Центра доступно в свободном режиме (в настоящее время), то этот режим в основном и использовался посетителями. Полномасштабный режим доступа к процессорным системам программных комплексов Центра, с возможностью решения гораздо более серьезных, чем в демоверсиях, задач, с необходимостью внесения абонентской платы через электронные платежные системы, использовался посетителями в существенно меньших масштабах.

На основе детального анализа статистических данных будет проведено планирование дальнейшего развития Web-ресурса — расширение контента новыми программными комплексами, модернизация действующих систем, улучшение дизайна и повышение комфорта пребывания клиентов в Центре.

Статистика браузеров посетителей

При разработке дизайна страниц сайта необходимо учитывать то важное обстоятельство, что страницы, разработанные для одного браузера, могут выглядеть по-другому при использовании других браузеров. Усилия по композиции размещения

контента (тексты, рисунки, окна ввода параметров в процессорные системы и т. п.), направленные на создание комфорта пребывания клиентов в Центре, могут быть затрачены напрасно. Весьма индивидуально выглядят напечатанные из разных браузеров страницы, а необходимость использования бумажных копий пока еще существует. Вследствие этого целесообразно при выполнении Интернет-проектов знать статистику данных, какие браузеры используются посетителями для просмотра Web-страниц, а также соответствующие количественные характеристики.

Приведем ряд статистических показателей за 12-месячный учетный период. Безусловно, в соответствии с реалиями настоящего времени, на первом месте используемых посетителями браузеров находится *Microsoft Internet Explorer* (MSIE) различных версий с высоким, 52 % от числа всех клиентских браузеров, показателем.

Второе и третье места занимают браузеры *Netscape* (20 %) и *Opera* (15 %) различных версий.

Отметим, что в прошлый учетный период с 1 апреля 2008 г. по 31 марта 2009 г. показатель использования браузеров MSIE был значительно выше и достигал значения 65 %, а следующие места занимали те же браузеры *Opera* (18 %) и *Netscape* (13 %).

Еще ранее, в статистический учетный период с 1 января по 31 декабря 2008 г. *top*-список браузеров был следующим: на первом месте также находился браузер MSIE с показателем 53 %, а следующие места занимали браузеры *Opera* (9 %) и *Netscape* (2 %).

Таким образом, следует констатировать, что наиболее распространенным браузером является MSIE. Этот браузер предпочитают более половины посетителей портала, и данный показатель весьма устойчив. Второе и третье места по распространенности среди посетителей портала попеременно делят браузеры *Opera* и *Netscape* с показателями не выше 20 % каждый.

Эта статистика учитывается в проекте — страницы сайта тестируются в браузере MSIE, а на браузерах *Opera* и *Netscape* дизайн создаваемых страниц постоянно контролируется.

Заключение

Статистический анализ является мощным средством исследования реального состояния проекта. Всестороннее изучение статистики с ее многообразными параметрами способствует правильному прогнозированию развития проекта и проведению своевременной и эффективной корректировки этого процесса в соответствии с разработанным планом.

Список литературы

1. **Облачные** войны только начинаются (операционная система Windows Azure): (<http://www.livebusiness.ru/news/4591/>).
2. **Тарнавский Г. А.** Первый международный конгресс по "облачным" технологиям — Cloud Computing Congress CCC-2010 // Информационные технологии. 2010. № 10. С. 77—78.
3. **Тарнавский Г. А., Чесноков С. С.** Компьютерное моделирование в Интернете: краткий обзор Web-ресурсов // Информационные технологии. 2010. № 5. С. 49—52.
4. **Тарнавский Г. А., Алиев А. В., Анищик В. С., Тарнавский А. Г., Жибинов С. Б., Чесноков С. С.** Информационные технологии и проблемы создания Центра компьютерного моделирования в Интернете // Информационные технологии. 2009. № 8. С. 68—73.
5. **Тарнавский Г. А.** Центр компьютерного моделирования в Интернете // Информационные ресурсы России. 2010. № 4. С. 36—40.
6. **Тарнавский Г. А.** Облачные вычисления: контент, инфраструктура и технологии организации информационных потоков Центра компьютерного моделирования SciShop.ru // Исследования в России. 2010. Т.13, № 001. С. 1—29. URL: <http://zhurnal.apc.relam.ru/articles/2010/001.pdf>
7. **Тарнавский Г. А.** Ударно-волновые режимы течения на входе в диффузор гиперзвукового прямоточного воздушно-реактивного двигателя: влияние высоты и скорости полета // Теплофизика высоких температур. 2005. Т. 43. № 1. С. 57—70.
8. **Тарнавский Г. А., Алиев А. В., Тарнавский А. Г.** Компьютерное моделирование в аэромеханике: программный комплекс "Поток-5" // Авиакосмическая техника и технология. 2007. № 4. С. 27—38.
9. **Алиев А. В., Тарнавский Г. А.** Иерархический SPH-метод для математического моделирования в гравитационной газовой динамике // Сиб. электронные математические заметки. 2007. Т. 4. С. 420—180.
10. **Тарнавский Г. А., Анищик В. С.** Инструментарий NanoMod компьютерной поддержки проектирования наноструктурированных полупроводниковых материалов // Вычислительные методы и программирование. 2009. Т. 10. Раздел 2. С. 34—50.

УДК 004.43

Ю. И. Рогозов, д-р техн. наук, проф., зав. каф.,
e-mail: rogozov@tsure.ru,
А. С. Свиридов, канд. техн. наук, доц.,
e-mail: sviridov@tsure.ru,
А. А. Дегтярев, ассистент,
e-mail: alexey.a.degtyarev@gmail.com,
Технологический институт
Южного Федерального университета, г. Таганрог

Анализ и перспективы развития разработки программного обеспечения

Проводится исторический анализ подходов к разработке программного обеспечения, выделяются основные скачки, дается краткая характеристика каждого подхода. Обосновывается актуальность развития и исследования подхода Model Based Software Engineering.

Ключевые слова: разработка программного обеспечения, языки программирования, ООП, CBSE, MDE, MSBE

Введение

Целью данной статьи является исторический анализ подходов разработки программного обеспечения, выявление основных тенденций и определение перспективных направлений исследований в области инженерии программного обеспечения. Использование автоматизации в процессе разработки программного обеспечения началось с компиляции и линковки (Linking), а продолжилось поддержкой создания и редактирования исходного кода программы, управления версиями, отладкой, верификацией, документированием и т. д.

Введение компиляторов привело к появлению языков программирования третьего поколения (3GL), которые снизили сложность проектирования программ за счет устранения необходимости программистам заботиться о многих технологических деталях (technology-specific details).

Стало возможным переносить программы на различные вычислительные машины практически без изменений и, что более важно, программировать с помощью понятий и конструкций, которые гораздо ближе к человеческому пониманию и к проблемной области. Однако, несмотря на много-

численные дополнительные улучшения, уровня абстракции, предоставляемого языками программирования третьего поколения, такими как C, FORTRAN, BASIC, оказалось недостаточно. В частности, базовые конструкции в этих языках оказывались слишком мелкими для предоставления прямого и ясного выражения более сложных и предметно-ориентированных понятий и отношений (например, для отображения архитектуры программного обеспечения или организационной структуры корпорации).

Дальнейшее повышение уровня абстракции спецификаций программного обеспечения осуществлялось путем использования формализмов более высокого порядка, таких как конечные автоматы или сущность — связь. Сегодня использование таких высокоуровневых формализмов — стандартная практика в области анализа и проектирования сложных программных систем. Автоматизация перевода высокоуровневых спецификаций программного обеспечения в код не достигла сопоставимого с компиляторами уровня. Сложившаяся ситуация расширила семантический разрыв между проектными спецификациями программного обеспечения и его реализацией. Как правило, спецификации выражаются с помощью формализмов более высокого уровня, а соответствующие им реализации — с помощью языков программирования третьего поколения.

В течение последних трех десятилетий в развитии языков программирования было множество успехов. Несмотря на это уровень предоставляемой абстракции существующими языками реализации (Java, C++ и C#) и, как следствие, выразительная сила незначительно больше, чем у самых ранних языков третьего поколения. Почти так же трудно различать высокоуровневые архитектурные формы и принципы построения программы, которая была написана на Java, как если бы ее написали на FORTRAN или COBOL. Очевидно, что такое расхождение в абстракции вызывает проблемы. В частности, это касается высокого уровня риска того, что при неформальном переводе высокоуровневых спецификаций в рабочие программы могут быть внесены ошибки, при которых конечная реализация не будет точно соответствовать задуманному проекту. Вместе с тем, игнорирование основных проблем реализации в высокоуровневых спецификациях может сделать систему неэффективной или даже неработоспособной. Сообщество разработчи-

ков программного обеспечения давно пришло к выводу, что для снижения к минимуму вероятности появления таких проблем сложные программные системы лучше разрабатывать с использованием инкрементального или итеративного процесса разработки, где анализ, проектирование и реализация должны выполняться либо параллельно, либо циклически друг за другом. При этом широкий семантический разрыв между абстракциями понятий и использование разных инструментов и методов вызывают дополнительные трудности для осуществления итераций или перехода между ними. Преодоление семантических разрывов в этих процессах с помощью автоматизации является очевидным выбором. К настоящему времени было сделано множество попыток внедрения такой автоматизации в разработку программного обеспечения, начиная с так называемых языков программирования четвертого поколения (4GL), инструментов автоматизированного проектирования программного обеспечения (CASE) и заканчивая текущими подходами разработки программного обеспечения, основанными на моделях (MBSE). Рассмотрим каждый из них в отдельности.

Анализ развития технологий, парадигм и подходов к разработке программного обеспечения

Термин "разработка программного обеспечения" (от англ. *soft ware development*) берет свое начало с 1940-х годов и был связан с непосредственным созданием программ для управления первыми ЭВМ (например ENIAC, 1946 г.). Тогда содержание этого термина определялось в первую очередь выбранной технологической подложкой — двоичной системой счисления. Впоследствии сформировалась научная дисциплина "инженерия программного обеспечения" (от англ. *soft ware engineering*), призывающая использовать подходы классических инженерных наук в создании программного обеспечения. В процессе развития инженерии программного обеспечения появлялись технологии, парадигмы, подходы, методы и т. д., которые оказывали друг на друга перекрестное влияние, именно поэтому между ними сложно провести четкую грань как в семантической плоскости, так и во временной. Несмотря на это мы попытались выделить основные серьезные скачки в развитии подходов, технологий, парадигм и методов и расположили их в условном хронологическом порядке. Такой исторический анализ позволит понять основной тренд в развитии подходов к разработке программного обеспечения, изучить влияние автоматизации на разработку программного обеспечения и рассмотреть перспективные области исследований.

Скачок первый: языки программирования третьего поколения. Для программирования первых ЭВМ использовали языки программирования первого и второго поколений (машинный код — 1GL и сбо-

рочные языки программирования (*assembler*) — 2GL), которые были жестко привязаны к архитектуре аппаратного обеспечения. Разработка программного обеспечения для информационных систем началась с широкого использования языков программирования третьего поколения, таких как FORTRAN (1957 г.), LISP (1958 г.), COBOL (1959 г.) и BASIC (1964 г.). Эти языки оказали сильное влияние на платформенную независимость программ и уровень абстракции, с которым работали программисты, что повлекло за собой массовое использование ЭВМ в различных областях человеческой деятельности.

Фактически произошло первое серьезное перераспределение обязанностей между участниками разработки программного обеспечения — программисты взяли на себя обязательства алгоритмистов, которые ранее выступали в роли связующего звена между специалистом-предметником и программистом, в задачи которого входило непосредственное кодирование разработанных ранее алгоритмов. Таким образом, языки 3GL — первый серьезный шаг на пути к автоматизации разработки программного обеспечения.

Скачок второй: парадигма структурного программирования. В конце 1960-х годов, когда Эдсгер Вйбе Дейкстра опубликовал свою статью в сообществе ACM (1968 г.), началась новая тенденция. В своей статье он утверждал, что оператор GOTO должен быть исключен из всех языков программирования высокого уровня. Несколько лет спустя в 1972 г. он вместе с другими соавторами публикует книгу "Структурное программирование". Интенсивное использование методологии структурного программирования в течение 20 лет и появление таких языков, как Паскаль (1970 г.), С (1972 г.) и Ада (1983 г.), делает этот период настоящим скачком в разработке программного обеспечения. Однако серьезного значения для автоматизации разработки программного обеспечения этот этап не представлял, он скорее стал фундаментом для дальнейшего развития автоматизации.

Скачок третий: языки программирования четвертого поколения. Языки программирования четвертого поколения (часто их называют 4GL) являются высокоуровневыми языками, специализированными для конкретной предметной области или цели. Одним из первых примеров таких языков является язык генерации отчетов Report Program Generator (RPG, International Business Mashines, 1964 г.). Он представляет собой декларативный язык, предназначенный для генерации отчетов из баз данных, и в свое время являлся весьма эффективным для поставленных перед ним задач, что значительно сокращало время разработки эквивалентных программ на языках программирования императивного стиля, таких как COBOL. Позже было разработано множество других языков 4GL, в частности, SQL

(Чемберлин и Бойс, 1974 г.) для запросов к базам данных, MATLAB (Math Works, 2008 г.) для математического моделирования и моделирования системной динамики, SPSS (IBM Company) — статистический пакет для социальных наук, которые доказали свою эффективность с течением времени. Возникшая в настоящее время концентрация внимания на предметно-ориентированном моделировании и языках (Domain Specific Modeling и Domain Specific Languages) является просто дальнейшим проявлением этой тенденции [1]. Главной особенностью языков четвертого поколения и в то же время их основным недостатком является высокоспециализированный характер.

Скачок четвертый: компонентно-ориентированная разработка программного обеспечения (CBSE — Component Based Software Engineering). Парадигма разработки программного обеспечения на основе компонентов начала формироваться в конце 1970-х годов как один из ответов на возникший тогда "кризис программного обеспечения" [2].

В 1980-х годах модульное программирование добавило такие понятия, как интерфейс и библиотека. Большинство структурных языков, упоминавшихся ранее, поддерживали модульное программирование или были расширены для его поддержки. Modula-2 (1978 г.) является ярким примером ориентации языка на новую тенденцию.

Объектно-ориентированную парадигму можно рассматривать как развитие первых идей, заложенных в CBSE [3]. Объектная ориентированность появилась еще в начале 1960-х годов как парадигма обучения, однако первый объектно-ориентированный язык Small Talk был выпущен в 1980 г. (стандартизированная версия ANSI Small Talk появилась в 1998 г.). Эта парадигма привлекла внимание сообщества разработчиков программного обеспечения в середине 1990-х годов и продолжает привлекать до сих пор.

Сегодня объектно-ориентированная парадигма является общепринятым распространенным способом разработки программного обеспечения для информационных систем. Вместе с успехом этой парадигмы появилось несколько новых языков (например, Java, 1995 г.), но более важно, что все существующие языки, поддерживающие парадигму структурного программирования, были адаптированы для поддержки какого-нибудь способа реализации объектно-ориентированного программирования (например C++, 1993 г. и Ada95, 1995 г.).

Скачок пятый: инструменты автоматизированного проектирования программного обеспечения CASE. Разработчики инструментов автоматизированного проектирования программного обеспечения (далее по тексту — CASE) были вдохновлены неоспоримым успехом систем автоматизированного проектирования (CAD), которые, в свою очередь, существенно подняли уровень автоматизации разработки

аппаратного обеспечения. В общем случае CASE-инструменты представляют собой ранние попытки прямого перевода высокоуровневых формализмов, используемых для анализа и проектирования программных систем в эквивалентный код на объектно-ориентированных языках третьего поколения (3GL) или специализированных языках четвертого поколения (4GL). Идея, которую они пропагандируют, кажется безукоризненной. Они не только способствовали более удобному процессу проведения этапов анализа и проектирования путем предоставления инструментов для создания различных диаграмм и спецификаций, но также в той или иной форме поддерживали автоматизированную генерацию программного кода или его частей. К сожалению, CASE-инструменты не достигли успеха, соразмерного с успехом CAD. Теперь в научных дискуссиях они часто упоминаются как пример еще одной технологии, которая обещала прорыв, но так и не смогла его осуществить.

Существует несколько причин "провала" CASE, которые стоит рассмотреть подробно, поскольку они могут служить уроками, которые могут быть учтены при последующих попытках автоматизации разработки программного обеспечения.

Итак, CASE-инструменты были направлены прежде всего на сокращение семантического разрыва между проектировщиками и программистами и по возможности на сокращение времени перехода от проекта к реализации. Одно из главных препятствий этому заключалось в качественном различии между проектировщиками и программистами.

Проектирование, особенно на ранних стадиях, безусловно лучше всего осуществлять в условиях, при которых идеи могут формироваться свободно от ограничений и формальностей. Следовательно, большинство языков для проектирования, которые поддерживались CASE-инструментами, как правило, поддерживали формализмы (нотации), которые были неточными и неформальными. Языки реализации, вместе с тем, обязательно должны быть однозначными и весьма формальными, так как они абсолютно точно переводятся в строго определенный машинный код. Таким образом, любой программный код, который генерируется из таких высокоуровневых формализмов, будет неполным и для запуска и компиляции потребует ручного дополнения программистом. Он исключит неоднозначности и дополнит недостающие части. Такой подход хорош, однако он нарушает формальные связи между кодом и спецификациями, из которых он был получен. Любое добавление или изменение кода может нарушить или даже начать противоречить первоначальным целям проектирования. В итоге автоматическое возвращение к высокоуровневым формализмам становится невозможным — это препятствует дальнейшей итеративной разработке.

Предпринимались многочисленные попытки обойти эту проблему, одна из них Round-trip engineering (RTE), которая заключалась в том, что сгенерированный и отредактированный программистом код подвергался обратной инженерии в соответствующие спецификации высокого уровня.

Однако наиболее частым результатом серии таких итераций является постепенное ухудшение высокоуровневых спецификаций путем превращения их во взаимно однозначное низкоуровневое графическое представление программного кода. В таком случае основное преимущество использования высокоуровневых спецификаций размывается. Кроме того, код, который создавался CASE-инструментами, часто был либо очень простым (что для разработчика представляется малозначимым), либо имел гораздо более низкое качество по сравнению с кодом, созданным вручную. Возможно, частично это было связано с тем, что практически отсутствовали теоретическое понимание и опыт того, каким образом генерировать код из графических формализмов.

Другая проблема CASE-инструментов была связана с переизбытком высокоуровневых формализмов, которые появились вместе с введением этих инструментов. Например, в начале 1990-х годов число опубликованных в научных сообществах языков анализа и проектирования приближалось к 100 [4]. Редко CASE-инструменты поддерживали более чем несколько семейств языков. Таким образом, CASE-инструменты не могли дать ничего особенного ни проектировщикам, ни разработчикам, однако они оказали несомненное влияние на развитие автоматизации разработки программного обеспечения и позволили наработать фундамент для начала следующего скачка.

Скачок шестой: разработка программного обеспечения, основанная на моделях (MBSE, MDE). Первоначально использование моделей для разработки программного обеспечения не концентрировалось вокруг идей автоматизации разработки. Одним из выдающихся языков, основанных на моделях, была нотация Сущность—Связь (ER) [5], предложенная П. П. Ченом в 1976 г. Концептуальное моделирование являлось активной темой исследования в течение 1980-х годов, тогда были опубликованы некоторые подходы, расширяющие ER (например EER, 1989 г. [6]). В 1991 г. благодаря вкладам Г. Буча (Booch Method), Дж. Рамбо (OMT) и И. Якобсона (OOSE) появились основы унифицированного языка моделирования (Unified Modeling Language, далее UML). В 1997 г., спустя двадцать один год, была представлена первая версия UML (UML 2.0 появился в 2005 г.). Сегодня UML используется для разработки программного обеспечения (здесь под термином разработки следует понимать и анализ, и проектирование, и даже реализацию) в качестве стандартного языка, основанного на моделях. ER и UML являются корнями MBSE (более часто встре-

чаются термины model-driven development (MDD) и model-driven engineering (MDE), однако мы считаем, что они вводят в заблуждение, реально модели не управляют разработкой, а лишь являются побочным продуктом разработки). Окончательно MBSE был осознан и начал формироваться с появлением Model Driven Architecture (MDA) [7] и интенсивного развития инициативных групп, которые воспринимали модели как основной элемент процесса разработки программного обеспечения. В настоящее время MBSE каждый год приобретает все большую популярность, но он все еще находится в инкубационном периоде.

Что касается автоматизации разработки программного обеспечения с использованием подходов MBSE, то по сути — это просто продолжение CASE-подхода. Тем не менее, попытки построить средства автоматизации разработки программного обеспечения на основе принципов MBSE существенно отличаются от построения CASE-систем, основанных только лишь на наличии высокоуровневых формализмов. В частности, был достигнут серьезный прогресс в базовых технологиях, где используется более мощная вычислительная техника (производительность, объем памяти), а также прогресс в области разработки языков моделирования (использование подходов метамоделирования) и методов автоматической генерации кода. Появились и получили широкое внедрение ряд ключевых промышленных стандартов, особенно Object Management Group's Unified Modeling Language (UML) (Object Management Group, 2007 г.), что привело к значительному сокращению избытка различных обозначений и языков моделирования высокого уровня. Именно поэтому подход автоматизации на основе MBSE является более жизнеспособным. Существует много публикаций по MBSE [8, 9, 10, 11].

Однако справедливо отметить, что MBSE по-прежнему далек от классической инженерной дисциплины, т. е. такой дисциплины, которая основана на хорошо понимаемой научной и технической базе. Интересно отметить, что сегодня ситуация с MBSE имеет значительное сходство с тем периодом, когда вместо языков ассемблера предлагалось использовать языки программирования третьего поколения. Большинство программистов тогда просто не верило в то, что компилятор сможет создавать ассемблерный код такого же качества, как и человек.

Особенности и актуальность MBSE

Как отметил В. Selic, разработка программного обеспечения на основе моделей (MBSE, MDD) может стать первым настоящим скачком поколений в разработке программного обеспечения, начиная с введения компиляторов [12]. Существует много дискуссий о платформи-независимых моделях (PIMs) и платформи-зависимых моделях (PSMs), метамоделировании, предметно-ориентированных

языках моделирования, трансформации моделей и т. д. По нашему мнению, основную идею MBSE можно сократить до двух фундаментальных понятий:

- повышение уровня абстракции, т. е. повышение уровня программных спецификаций еще дальше от лежащих в основе технологий реализации (относящихся к традиционным языкам программирования);
- повышение степени автоматизации, которая используется для устранения разрыва между проектными спецификациями и соответствующими реализациями.

В большинстве инженерных практик термин "модель" используется для обозначения абстрактного представления некоторых конкретных инженерных или других артефактов, т. е. абстракция уходит от неинтересных деталей. Модель может быть математической, макетной или иметь какой-либо другой вид, но во всех случаях отличается от сущностей реального мира, которые она представляет. Однако в контексте MBSE "модель" часто используется как общий термин для обозначения любых спецификаций, которые выражаются с помощью формализмов более высокого уровня, чем языки программирования, независимо от того, является ли "модель" абстракцией, которая опускает подробности, или полноценной спецификацией реализации, из которой может быть автоматически сгенерирована полноценная исполняемая программа. Эту своеобразную практику можно отнести к уникальной природе MBSE, в которой окончательный артефакт разработки, в принципе, может быть результатом ряда инкрементальных уточнений набора последовательных высокоуровневых спецификаций. В ходе этого процесса непрерывно можно использовать одни и те же языки, инструменты, фреймворки, методы и опыт. Это позволит избежать качественных скачков, которые характеризуют практически все другие формы инженерной деятельности. При таком процессе, если автоматизация будет достаточно формально и правильно организована, можно надеяться на гораздо большую вероятность сохранения первоначальных целей проектирования. Стоит отметить, что на ранних стадиях этого непрерывного процесса полезно иметь относительно низкий уровень формальности и точности, что позволит более свободно выражать проектные идеи. По мере выполнения процесса должны соответственно увеличиваться степень формальности и проверка согласованности, пока в финальной стадии

степень формальности не станет связана с языками программирования. Требование поддерживать постепенное повышение уровня формальности является очень важной особенностью, которую следует учитывать при выборе языков моделирования.

Исторический анализ и выводы

Разработка программного обеспечения имеет очень изменчивую природу, несмотря на то, что это очень "молодая" дисциплина. Каждый новый скачок в разработке программного обеспечения стремится повысить уровень абстракции, что соответствует развитию методологий человеческого мышления и соответственно повышению независимости программного обеспечения от платформ. Концептуально соответствие рассмотренных выше скачков можно представить рис. 1. На рис. 2 приведен примерный график развития шести скачков в разработке программного обеспечения, рассмотренных в предыдущем разделе. Градиент цвета на каждой линии отражает интенсивность использования подхода в практике программирования.

Анализируя линии жизни скачков, можно выявить следующие закономерности (рис. 2):

- все они имеют инкубационный период. Основную часть этого периода занимают исследования, и именно здесь появляются новые идеи и

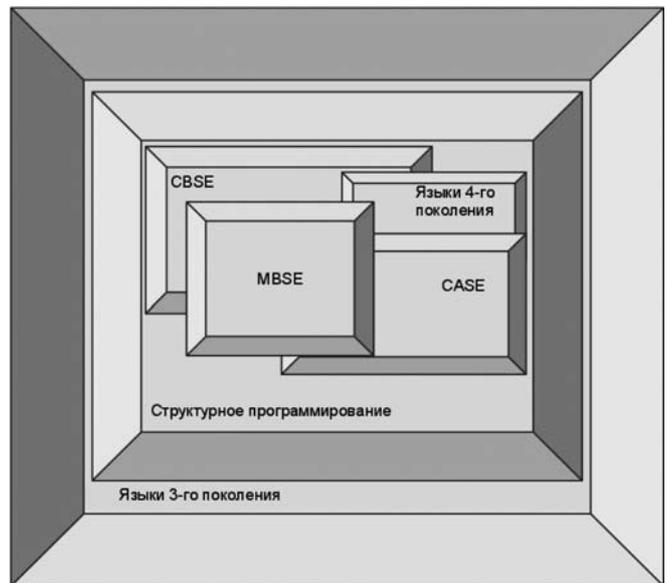


Рис. 1. Развитие подходов разработки

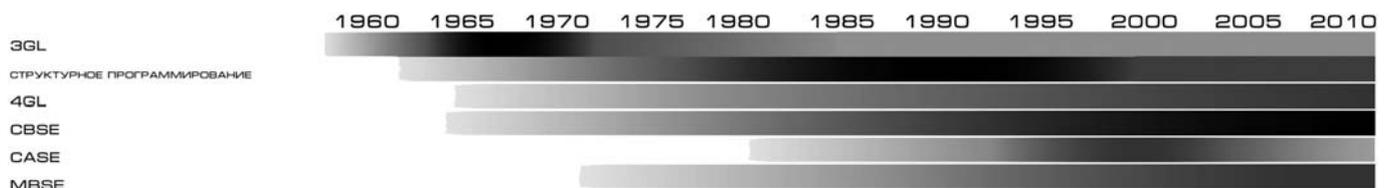


Рис. 2. Линии жизни подходов к разработке программного обеспечения

- языки; стоит также отметить, что в каждом новом скачке инкубационный период увеличивается;
- почти все они имеют период интенсивного использования, который, как правило, тоже больше в каждом новом скачке;
 - все они, как правило, имеют переходной период, где наиболее успешные языки и инструменты адаптируются для поддержки идей следующего скачка.

Самая главная закономерность, которая проявилась в результате исторического анализа, — это то, что каждый следующий скачок поглощает наиболее успешные идеи и технологии с предыдущего скачка и добавляет к ним какие-либо новые идеи, что обеспечивает очередное повышение уровня абстракции. Иными словами, речь идет о том, что методы MBSE сейчас используют в своей основе какой-либо метод CBSE (как правило, ООП), который, в свою очередь, использует идеи и технологии структурного программирования, а тот — технологии и методы языков программирования третьего поколения.

В этом историческом обзоре и анализе мы попытались показать, что подход автоматизации на основе MBSE является реальным кандидатом, ко-

торый обеспечит новый скачок в разработке программного обеспечения, и что MBSE в ближайшем будущем будет становиться все более актуальным.

Список литературы

1. Greenfield J., Mernik M., Heering J., Sloane M. When and how to develop domain-specific languages // Software Factories. Wiley, Indianapolis (2004): ACM Computing Surveys 37 (4). 2005. P. 316—344.
2. Dijkstra, E. W. The Humble Programmer // ACM Turing Award Lectures, 1972.
3. Cox B. J. Object oriented programming: an evolutionary approach. Addison-Wesley, Reading, MA, 1986.
4. Graham I. Object-Oriented Methods. Addison-Wesley, London, 2001.
5. Chen P. P. The Entity-Relationship Model: Toward a Unified View of Data // ACM Trans. on Database Systems. 1976. Vol. 1. P. 9—36.
6. Embley D., Ling W., Tok Wang. Synergistic Database Design with an Extended Entity-Relationship Model // ER. 1989. P. 111—128.
7. MDAGuideVersion 1.0.1 // Object Management Group URL: <http://www.omg.org/docs/omg/03-06-01.pdf> (23.03.2011).
8. MDAGuide, v.1.0.5. // Object Management Group. URL: <http://www.omg.org/docs/omg/2003-06-01.pdf> (23.03.2011).
9. Frankel D. Model Driven Architecture — Applying MDA to Enterprise Computing. OMG Press, Indianapolis, 2003.
10. Greenfield J. et al. Software Factories. Wiley, Indianapolis, 2004.
11. Mellor S. et al. MDA Distilled — Principles of Model-Driven Architecture. Addison, Wesley, Boston, 2004.
12. Selic B. The Pragmatics of Model-Driven Development // IEEE Software. 2003. Vol. 20. P. 19—25.

УДК 004.4'413

В. В. Лидовский,

Российский государственный технологический университет им. К. Э. Циолковского — "МАТИ",
e-mail: lidw@rambler.ru

Анализ LR-разбора для 18 языков программирования

Исследуется актуальность использования канонических LR(1)-таблиц для разработки компиляторов. Приводятся результаты тестирования грамматик 18 широко используемых языков программирования. Делается вывод об полезности поддержки LR(1)-разбора генераторами компиляторов.

Ключевые слова: компиляторы, восходящий (LR) разбор, канонические LR-таблицы, LALR, Бизон, Як

Как и LALR-разбор, LR-разбор имеет линейную временную сложность. LR-разбор имеет несколько теоретических преимуществ. Использование LR(1)-разбора вместо LALR(1)-разбора может устранить некоторые конфликты типа свертка-свертка, которые, как отмечается в работе [8], "обозначают серьезную проблему" при разработке компилятора:

LR(1)-разбор успешно справляется с ситуациями, порождающими "таинственные конфликты свертка-свертка" [1] при LALR(1)-разборе. Кроме того, использование LR(1)-разбора позволяет наиболее точно специфицировать ошибки при компиляции. При LALR(1)-разборе диагностика ошибок менее точна, так как многие ошибочные ситуации объединяются.

После опубликования Д. Кнудом в 1965 г. работы [2], определяющей LR(k)-алгоритм, прошло более 10 лет, прежде чем его упрощенный вариант — LALR(1), чрезвычайно удачно реализованный С. Джонсоном в 1972 г. в программе Як (yacc), стал вытеснять прежде преобладавшие методы разбора сверху-вниз. Необходимость упрощения вытекала из аппаратных ограничений вычислительных машин того времени. Еще в 1985 г. в фундаментальном труде [7] отмечается, что "построение полной системы LR(1)-множеств пунктов требует слишком много памяти и времени, чтобы использоваться на практике".

Автором и М. Молокановым на кафедре "Моделирование систем и информационные технологии" МАТИ-РГТУ им. К. Э. Циолковского было проведено соответствующее исследование в течение семестра. На языке программирования Си++ со стандартной библиотекой была реализована ком-

пьютерная программа lr-lalr-test, которая по заданной грамматике строит соответствующую ей полную каноническую систему LR(1)-множеств пунктов, а также полную систему LALR(1)-множеств пунктов. Последнее построение позволяет, в частности, верифицировать результаты программой Бизон. Программа также вычисляет число терминалов, метасимволов, правил, LR- и LALR-множеств пунктов и другие численные характеристики грамматики. Выбор языка программирования был обусловлен прежде всего необходимостью проведения больших объемов расчетов за приемлемое время и поддержкой быстрой работы с множествами. Языки Лисп и Пролог, а также Перл, Питон, Рубин и некоторые другие, имеющие необходимые средства, показали значительно меньшую скорость работы. Использовалась система программирования GNU Си++: компилятор, отладчик, профайлер и компоновщик. Алгоритм для построения канонической системы множеств пунктов взят из работы [7].

Для анализа использовались следующие языки в их свободно распространяемых вариантах с открытыми текстами исходников (далее следует список языков с указанием названия, года версии исходников, разработчика, сетевого адреса, примечаний):

- Бизон — bison (2008, FSF, <http://www.gnu.org/software/bison/>, GNU вариант синтаксического сканера Як);
- Биси — bc (1997, FSF, <http://www.gnu.org/software/bc/>, консольный калькулятор произвольной точности);
- Бэш — bash (2009, FSF — Free Software Foundation, <http://tiswww.case.edu/php/chet/bash/bashtop.html>, язык стандартной оболочки Linux);
- Майэскьюэль — mysql (2008, MySQL AB и Sun Microsystems, <http://www.mysql.com/>, вариант Эскьюэль — SQL);
- Обджектив Си — Objective C (2001, FSF, <http://gcc.gnu.org/>, GNU вариант похожего на Си языка фирмы Apple);
- Оок — awk (2009, FSF, <http://www.gnu.org/software/gawk/>, gawk — GNU вариант);
- Паскаль (2006, FSF, <http://www.gnu-pascal.de/gpc/h-index.html>, gpc — GNU вариант расширенного Паскаля);
- Перл (2008, Larry Wall and others, <http://www.perl.org/>);
- Питон — python (2008, PSF — Python Software Foundation, <http://www.python.org>, исходники из расширенных НФБН, принятой в PSF, были преобразованы к обычным НФБН);
- Постгрескьюэль — PostgreSQL (2010, Global Development Group, <http://www.postgresql.org/>, вариант Эскьюэль);
- ПХП — php (2010, Zend Technologies, <http://www.php.net>);

- Рубин — ruby (2007, Юкихиро Мацумото — Yukihiro Matsumoto, <http://www.ruby-lang.org/en/>);
- Си (2001, FSF, <http://gcc.gnu.org/>, gcc — GNU вариант Си);
- Си++ (2001, FSF, <http://gcc.gnu.org/>, g++ — GNU вариант Си++);
- Флекс — flex (1990, The Regents of the University of California, <http://flex.sourceforge.net/>, GNU вариант лексического сканера Лекс — lex);
- Хок — hoc (1984, Керниган и Пайк, <http://litwr.atstspace.com>, расширение описанного в [8] программируемого консольного калькулятора);
- Ява (2001, FSF, <http://gcc.gnu.org/>, GNU вариант языка Ява);
- PL/pgSQL (2010, Global Development Group, <http://www.postgresql.org/>, встроенный в Постгрескьюэль процедурный язык).

Грамматика для программы lr-lalr-test задается файлом в особом формате, получаемом из исходного файла в формате Як/Бизон обработкой специально разработанной программой-препроцессором. Цель препроцессора — получить "чистую" грамматику, в частности, без информации о приоритетах знаков и правил. Если рассматривать этот препроцессор, написанный на языках bash и awk, как часть конвейерной системы тестирования, то можно считать, что форматом грамматики является формат для программ Як/Бизон. Полученные результаты приводятся в таблице.

Число LALR-состояний получается на единицу меньшим, чем у программ Як/Бизон, вследствие отсутствия избыточного финального сдвига по особому конечному символу входной цепочки. Число

Язык	(1)	(2)	(3)	(4)	(5)	(6)	
Бэш	59	38 (0)	167	680	3684	343	0,84
Бизон	56	33 (3)	105	267	239	142	0,04
Биси	49	35 (13)	107	324	1149	184	0,41
Майэскьюэль	588	837 (176)	2377	6937	4823743	4076	?
Обджектив Си	87	237 (65)	589	1814	4353	991	13,57
Оок	66	56 (5)	165	526	3217	307	4,98
Паскаль	138	294 (65)	797	2394	39355	1329	20,22
Перл	89	66 (2)	209	727	3600	418	18,37
Питон	85	174 (83)	333	876	4758	529	1,62
Постгрескьюэль	430	515 (0)	2090	6926	873573	3837	2426,76
ПХП	153	181 (68)	463	1512	10986	892	33,4
Рубин	148	170 (32)	561	1730	180954	970	306,55
Си	87	167 (40)	426	1357	2888	728	7,29
Си++	112	294 (57)	918	3030	31683	1789	109,25
Флекс	68	27 (0)	97	275	232	139	0,05
Хок	40	17 (2)	63	210	692	119	0,93
Ява	110	163 (12)	504	1721	5594	774	17,01
PL/pgSQL	99	78 (2)	179	446	684	249	0,26

(1) — число терминалов; (2) — число метасимволов (неявных); (3) — число правил; (4) — вес грамматики; (5) — число LR(1)/LALR(1)-множеств; (6) — время, с.

терминалов на два, а нетерминалов на $n + 1$, где n — число неявных пустых правил (правил, порождаемых семантическими действиями между символами правил грамматики), больше, чем в исходной грамматике. Добавленные терминалы — это уже упомянутый конечный символ входной цепочки и символ-ошибка, а добавленный нетерминал — это начальный символ расширенной грамматики.

Очевидно, что общее число вариантов LR(1)-пунктов не превосходит wt , где w — это суммарное число символов, кроме пустых, во всех правилах — вес грамматики, а t — число терминалов. Следовательно, общее число LR(1)-состояний не превосходит 2^{wt} . Вследствие того, что в LALR-пунктах FIRST-часть пунктов теряет различающее значение, максимум числа LALR-пунктов в t раз меньше и, соответственно, максимум общего числа подмножеств всех LALR-пунктов — 2^w .

Время работы по сравнению с программой Бизон увеличилось значительно — на несколько порядков — соответственно росту числа множеств-состояний. Однако время построения LR(1)-системы вполне сопоставимо со временем компиляции файлов исходников языков программирования в исполняемые файлы. Кроме того, возможна лучшая оптимизация кода. Как уже отмечалось, разница в количестве LALR-и LR-состояний, в общем случае, определяется экспоненциальной зависимостью, что делает LR-разбор в крайних ситуациях фундаментально неэффективным. Эта общая теоретическая неэффективность практически проявилась только к языку Майэскьюэль, для которого даже 16 Гбайт оперативной памяти оказались недостаточными, и частично к языкам Рубин и Постгрескьюэль, потребовавшим соответственно 1 и 4 Гбайт для вычислений без привлечения медленной виртуальной (дисковой) памяти. Поэтому можно предположить, что *использование LR-разбора как одной из опций восходящего анализа является вполне приемлемым для ресурсов существующих компьютеров во многих случаях*. Кроме того, необязательно использовать именно канонические таблицы. Существуют несколько алгоритмов [3—6, 9], дающих возможность получать только существенные для работы анализатора состояния, что позволяет значительно сократить общее число LR-состояний и намного увеличить скорость работы программы. Хотя в этом случае *будет теряться часть информации об ошибках разбора*.

Помимо использования программы Бизон результаты верифицировались LR-сканерами Msta (Msta) и Нуасс. Для Майэскьюэль результат удалось получить только Mстой.

Учитывая идентичность алгоритмов LALR- и LR-разборов (они отличаются только наполнением таблиц) можно естественно интегрировать LR-разбор в существующие программы для генерации компиляторов, права копирования на которые позволя-

ют проводить такую разработку. Можно, например, добавить как опцию LR-разбор программам Як или Бизон. В последнем случае можно использовать возможность подключения к проекту Бизон в отдельной *svn*-ветке.

Уже существует несколько программ для LR(1)-разбора (CSP, Dragon, Нуасс, Lisa, Yooparse, Whale, ...), но они либо вводят собственный не вполне совместимый с Бизон синтаксис для определения грамматик, либо не реализуют всех возможностей программы Бизон, что, учитывая широкую распространенность исходников и прочих материалов для Як/Бизон, не вполне удобно. В частности, при тестировании программы Yooparse были обнаружены ошибки в построенной системе множеств пунктов для языка Перл — точнее не всегда вычисляются правильно FIRST-части пунктов. Новая программа Нуасс показала не полную совместимость с форматом грамматики программ Як/Бизон и слишком расточительное обращение с памятью — она не смогла вычислить каноническую таблицу LR-анализа на компьютере с 8 Гбайт памяти для языка Рубин. Кроме того, часть LR(1)-анализаторов — это далее не сопровождаемые проекты, использование которых проблематично.

Исходники программы lr-lalr-test и документация к ней доступны по адресу <http://litwr.atspace.com> — они распространяются на условиях лицензии GNU GPL (<http://www.gnu.org/copyleft/gpl.html>).

Среди побочных результатов исследования было обнаружено значительное расхождение с работой [10] по численным характеристикам синтаксиса, это свидетельствует о том, что грамматики для разных версий языка, а также грамматики, написанные для генерации синтаксических анализаторов, и грамматики, теоретически определяющие язык, могут весьма значительно различаться по этим характеристикам.

Список литературы

1. **Donnelly C., Stallman R.** Bison: 2 April 2009, version 2.4.1 — Free Software Foundation. URL: <http://www.gnu.org>.174 p.
2. **Knuth D.** On the translation of languages from left to right // Information and control. 1965. N 8. P. 607—639.
3. **Pager D.** A Practical General Method for Constructing LR(k) Parsers // Acta Informatica. 1977. N 7. P. 249—268.
4. **Pager D.** Eliminating Unit Productions from LR Parsers // Acta Informatica. 1977. N 9. P. 31—59.
5. **Pager D.** The Lane-Tracing Algorithm for Constructing LR(k) Parsers and Ways of Enhancing Its Efficiency // Information Sciences. 1977. N 12. P. 19—42.
6. **Pager D.** The lane tracing algorithm for constructing LR(k) parsers // Proc. of the fifth annual ACM symposium on Theory of computing. 1973. P. 172—181.
7. **Ахо А., Сети Р., Ульман Д.** Компиляторы: принципы, технологии и инструменты: пер. с англ. М.: Вильямс, 2003. 768 с.
8. **Керниган Б., Пайк Р.** UNIX — универсальная среда программирования: пер. с англ. М.: Финансы и статистика, 1992. 304 с.
9. **Макаров В. Н.** Практичный метод оптимизации LR(1)-анализаторов // Программирование. 1988. № 3. С. 38—48.
10. **Свердлов С.** Арифметика синтаксиса // PC Week. 1998. № 42—43 (166—167). С. 84—87.

УДК 004.91-004.93

Н. В. Дербенев, аспирант, ассистент,
В. О. Толчеев, д-р техн. наук, проф.,
e-mail: tolcheev-vo@mail.ru,
Московский энергетический институт
(технический университет)

Выявление нечетких дубликатов в наукометрическом анализе

Рассматривается проблема выявления нечетких дубликатов в научной периодике. Обосновывается необходимость разработки специализированной процедуры идентификации нечетких дубликатов и целесообразность ее использования при проведении наукометрических исследований. На сформированных выборках определяются пороговые значения, разделяющие документы на уникальные и нечеткие дубликаты.

Ключевые слова: нечеткие дубликаты, наукометрический анализ, библиографические текстовые документы, расстояние Джаро-Винклера, коэффициенты ассоциативности, метод шинглов, меры и порог близости

Введение

В последнее время научное сообщество уделяет повышенное внимание разработке методов корректной оценки публикационной активности ученых в целях определения их научного авторитета и вклада в развитие предметной области. В центре дискуссии находятся вопросы создания и применения информационно-аналитической системы "Российский индекс научного цитирования" (ИАС "РИНЦ") [1, 2, 3].

При построении ИАС "РИНЦ" компания-разработчик "Научная электронная библиотека eLIBRARY.RU" реализовала большинство из стандартных, хорошо себя зарекомендовавших наукометрических индикаторов, которые успешно используются в зарубежных аналогах (Web of Science, Scopus). Функциональные возможности ИАС "РИНЦ" позволяют наряду с количественными показателями продуктивности автора анализировать также характеристики, отражающие "качественный" уровень публикаций (прежде всего, коэффициент цитирования, коэффициент Хирша, импакт-фактор издания) [4].

Вместе с тем в современной наукометрии при оценке как количественных, так и "качественных" результатов публикационной активности ученого не используются методы выявления дубликатов

(полностью совпадающих статей) или нечетких дубликатов (почти дубликатов). Игнорирование существования этой проблемы вряд ли обосновано. В условиях, когда получение грантов, карьерный рост и зарплата в академических и образовательных институтах непосредственно связаны с количеством публикаций, нельзя исключить возможности появления работ автора (группы авторов) с практически одинаковым содержанием в различных журналах. Одним из примеров "важности" наличия большого числа статей является конкурс РФФИ, где научный руководитель проекта должен указать пять журнальных публикаций за последние три года (при этом в случае получения гранта одной из основных форм отчетности опять же является число печатных работ). Другой не менее характерный пример — публикация статей с небольшими изменениями в течение нескольких лет перед защитой докторской диссертации для соответствия требованиям ВАК.

В последнее время появились новые негативные моменты: покупка статей и монографий, написанных другими авторами и являющихся чаще всего компиляцией результатов предыдущих работ, стремительный рост числа печатных изданий при весьма незначительном числе ученых, активно участвующих в НИОКР. Так, в ИАС "РИНЦ" отслеживается около 2500 российских научных журналов, причем, по мнению разработчиков, только около 1500 изданий имеют приемлемые наукометрические показатели и должны быть включены в РИНЦ [1]. Рост числа журнальных публикаций происходит на фоне резкого снижения числа ученых, занятых научной деятельностью. По оценкам специалистов НИУ-ВШЭ, лишь 16 % преподавателей вузов ведут реальную исследовательскую работу [5]. В сложившейся ситуации можно предположить, что одним из источников наполнения многочисленных журнальных номеров становятся никем неотслеживаемые клонированные статьи.

При несовершенстве инструментария наукометрического мониторинга тиражирование статей-клонов при очевидной простоте реализации имеет ничтожную вероятность обнаружения. Более того, в российских научных организациях, как представляется, отношение к размножению одних и тех же результатов и появлению дубликатов значительно более лояльное, чем к плагиату или некорректному и неполному цитированию.

Необходимо отметить, что за рубежом при распределении грантов и выдвижении ученого на на-

учные и профессорские должности главным образом анализируются не количественные, а "качественные" показатели. В последнее время одной из активно обсуждаемых тем стали более общие вопросы научной этики, в частности, борьба с фальсификацией результатов исследований, описанием фиктивных экспериментов, подтасовкой данных [6, 7]. Зарубежное научное сообщество обеспокоено итогами социологических опросов ученых, согласно которым около 1 % респондентов не отрицали, что в своих работах хотя бы раз заимствовали чужую информацию и (или) давали субъективную интерпретацию фактов. По мнению экспертов, в некоторых предметных областях (например, медицине, биологии, генной инженерии) и странах (прежде всего Китае) этот показатель значительно выше [8, 2]. Наибольший общественный резонанс получили два случая подделки результатов: целенаправленное искажение данных для подтверждения гипотезы о глобальном потеплении (так называемый Климатгейт) и фальсификация итогов научных исследований южнокорейских генетиков по созданию эмбриональных стволовых клеток человека.

Анализ иностранной литературы показал, что проводимые работы в области обработки текстовой информации не ограничиваются вопросами определения одинаковых Web-страниц при поиске информации, исключения совпадающих документов из хранилищ информации, выявления плагиата [9, 10]. За рубежом особое внимание уделяется созданию систем автоматизированного мониторинга документального потока в целях обнаружения "неожиданных", аномальных результатов, которые могут свидетельствовать о получении принципиально новых знаний или наличии грубых ошибок и преднамеренных искажений [11].

Действительно, представляется целесообразным и своевременным при анализе научной периодики выявлять аномальные события [12]. В контексте данной статьи под аномальными событиями понимаются нечеткие дубликаты.

Идентификация нечетких дубликатов научных публикаций — крайне деликатная задача, так как в подавляющем большинстве случаев появление статей одного и того же автора (группы авторов) с близкими названиями объясняется спецификой и длительностью проведения НИОКР, терминологией предметной области. Поэтому для выработки окончательного суждения по поводу статей, у которых программно-алгоритмические средства выявили высокую степень подобия, необходимо проводить дополнительную экспертизу с привлечением специалистов. Например, если статья автора, представленная в редакцию журнала, вызвала "тревожное" срабатывание специализированной процедуры, то рецензенту, назначенному редколлегией, целесообразно ознакомиться с полным текстом публика-

ций, которые могут являться нечеткими дубликатами, и проанализировать их смысловую близость.

На практике для обнаружения нечетких дубликатов из потока научной периодики достаточно дополнить ИАС "РИНЦ" программно-алгоритмическим модулем, позволяющим анализировать публикации автора на совпадения. Такой модуль можно также создать в ходе реализации нового проекта Минобрнауки "Разработка ИАС статистического анализа библиографических показателей науки на основе РИНЦ" [13].

Главными вопросами, на которые необходимо ответить при проведении идентификации дубликатов, являются: какие методы лучше других подходят для решения поставленной задачи; как сформировать выборки для исследований и как выбрать порог, корректно отделяющий оригинальные публикации от возможных дубликатов.

Обзор и систематизация методов выявления нечетких дубликатов

В данной работе рассматриваются вопросы выявления дубликатов (нечетких дубликатов) научных статей по их библиографическим описаниям (автор (соавторы), название, аннотация, ключевые слова, место и время издания и т. п.). Именно библиографическая информация имеется в открытом доступе в ИАС "РИНЦ". По мере наращивания возможностей ИАС можно будет распространить анализ на полнотекстовые документы.

В качестве математической модели библиографического документа используется векторная модель [14]:

$$\mathbf{X}_j = [x_j^{(1)}, \dots, x_j^{(i)}, \dots, x_j^{(M)}]^T, \quad (1)$$

где $x_j^{(i)}$ — вес термина i в документе j ($j = 1, \dots, N$; N — число документов в выборке; $i = 1, \dots, M$; M — число терминов после удаления служебных слов).

Значение $x_j^{(i)}$ будет существенно зависеть от используемого метода анализа текстовых данных (в некоторых алгоритмах $x_j^{(i)}$ обозначает не только вес термина, но и позицию слова в документе). Модель, описываемая выражением (1), задает лексическое содержание документа в виде набора несвязанных между собой слов [14, 15].

В настоящее время существуют различные трактовки понятий "дубликат" и "нечеткий дубликат", их применение во многом зависит от цели и задач исследования. Введем следующие определения.

Дубликат текстового документа в узком смысле — это документ идентичный по лексическому содержанию исходному (определяется с помощью автоматизированных процедур обработки и анализа библиографических (или полных) текстов, представленных в виде векторной модели).

Дубликат текстового документа в широком смысле — это документ, идентичный по *смысловому* содержанию исходному (определяется экспертно по полнотекстовым документам).

Выявление дубликатов в узком смысле проводится на основе анализа терминов документов, поэтому возможно появление ситуаций, при которых тексты, не совпадая полностью, будут очень похожи друг на друга. Такие документы, имеющие близкое лексическое (терминологическое) содержание, называются **нечеткими (неполными) дубликатами**. Фактически отнесение (или не отнесение) документа к категории нечетких дубликатов определяется порогом близости, значение которого определяется экспериментальным или экспертным способом.

В настоящее время практически отсутствуют исследования, направленные на разработку и применение в наукометрическом анализе процедур обнаружения нечетких дубликатов научных публикаций. Тем не менее в литературе достаточно широко представлены работы, проводимые в смежных областях (выявление плагиата и аномалий, создание цифровых библиотек и больших хранилищ данных, поиск информации) и предлагающие ряд интересных алгоритмов, которые целесообразно применять для решения задачи разделения публикаций на оригинальные и неоригинальные [9, 10, 11, 14, 15, 17, 18, 19].

Систематизация наиболее эффективных подходов идентификации нечетких дубликатов позволила разнести их по четырем основным группам [16].

1. Специализированные расстояния. К этой группе относятся *расстояния Хэмминга, Левенштейна, Дамерау—Левенштейна, Джаро—Винклера* и другие [17, 18]. В последнее время чаще всего вместо простых подходов (например расстояния Левенштейна) используются более комплексные меры, учитывающие различные особенности сравниваемых текстов. К таким мерам прежде всего относятся расстояния *Джаро* (d_J) и *Джаро—Винклера* (d_W), которые можно рассчитывать следующим образом:

$$d_J = \frac{1}{3} \left(\frac{m}{s_1} + \frac{m}{s_2} + \frac{m-t}{m} \right); \quad (2)$$

$$d_W = d_J + (lp(1 - d_J)), \quad (3)$$

где m — число соответствующих символов (символы считаются соответствующими, если они равны и находятся не далее, чем $(0,5\max\{s_1, s_2\} - 1)$ друг от друга); t — число перестановок (вычисляется как число соответствующих символов, расположенных в различном порядке, деленное на 2); s_1, s_2 — длины сравниваемых строк; l — число общих начальных символов (не больше четырех); p — масштабный коэффициент.

Указанные меры детально изучены и хорошо себя проявили для сравнения строк малой длины. Однако при анализе достаточно больших текстов они те-

ряют свою эффективность вследствие большой вычислительной сложности и невысокой точности.

2. Метод шинглов. Данный метод основан на расчете контрольных сумм (шинглов) для каждой последовательности слов в тексте [19]. Последовательности имеют фиксированную длину, которая может настраиваться в зависимости от длины документа. Для определения схожести документов обычно используют коэффициент Джаккарда:

$$C(\mathbf{X}, \mathbf{Y}) = \frac{|S(\mathbf{X} \cap S(\mathbf{Y}))|}{|S(\mathbf{X} \cup S(\mathbf{Y}))|}, \quad (4)$$

где $S(\mathbf{X})$ — набор шинглов документа \mathbf{X} ; $S(\mathbf{Y})$ — набор шинглов документа \mathbf{Y} , документы представляются в виде векторной модели (1).

С помощью такого подхода можно успешно определять полнотекстовые нечеткие дубликаты, в которых имеется большое число идентичных строк. К сожалению, метод шинглов несколько хуже работает с библиографическими описаниями ввиду их малого размера. Кроме того, он достаточно сложен в реализации и требует настройки ряда параметров.

3. Методы на основе расчета весов терминов. Эта группа включает в себя множество различных процедур определения информативных терминов, вычисления их веса и сравнения документов, описанных моделью (1), с помощью подходящих мер близости. В отличие от специализированных расстояний, ориентированных на учет перестановок в тексте, процедуры данной группы направлены на анализ частот встречаемости терминов в документе и выборке.

Обычно на практике используется *tf-idf-взвешивание (term frequencies-inverse document frequencies)*, согласно которому вес слова i в документе j зависит от числа вхождений слова в данный текст и общего числа документов в выборке, в которых также содержится этот термин [14]:

$$x_j^{(i)} = f_{ij} \log\left(\frac{N}{N_i}\right).$$

Здесь f_{ij} — частота слова i в документе j ; N — число документов в выборке; N_i — общее число документов, содержащих слово i .

Одним из наиболее распространенных способов определения близости между текстами является косинусоидальная мера, имеющая ясную геометрическую интерпретацию (угол между векторами, заданными терминами документов) [14]:

$$d(\mathbf{X}, \mathbf{Y}) = \cos(\mathbf{X}, \mathbf{Y}) = \frac{\sum_{i=1}^M x^{(i)} y^{(i)}}{\sqrt{\sum_{i=1}^M (x^{(i)})^2 \sum_{i=1}^M (y^{(i)})^2}}. \quad (5)$$

Методы на основе расчета весов терминов имеют высокую точность на документах среднего и большого размера. В случае, если тексты содержат несколько предложений, что характерно для библиографических описаний, эффективность алгоритмов сильно падает вследствие практически равных весов всех слов в публикации.

4. Коэффициенты ассоциативности (сопряженности). Данный подход основан на расчете различных коэффициентов из логической двумерной таблицы ассоциативности пары текстовых документов [17, 12].

Из большого числа разработанных коэффициентов ассоциативности для выявления нечетких дубликатов чаще всего используется коэффициент Джаккарда. В обозначениях таблицы ассоциативности формула (4) примет вид

$$J = \frac{A}{A + B + C}, \quad (6)$$

где A — число совпавших терминов в двух документах X и Y ; B — число терминов, имеющих в X и отсутствующих в Y ; C — число терминов, имеющих в Y и отсутствующих в X .

Данный подход прост в реализации и обеспечивает легко интерпретируемые результаты, при этом он имеет достаточно высокую точность.

Формирование выборок и определение порога близости на основе экспериментальных исследований

Настройка порогового значения, разделяющего уникальные и неуникальные публикации, является наименее формализованной частью процедуры выявления нечетких дубликатов. На значение порога существенное влияние оказывает специфика предметной области, используемые массивы текстовой информации, квалификация специалистов, участвующих в экспертизе результатов.

Основная проблема состоит в том, что весьма сложно сформировать выборку, содержащую достаточно большое число неоригинальных работ. Это связано с тем, что появление статей-клонов в научной периодике относится к аномальным событиям, отслеживанию которых в настоящее время не уделяется должного внимания. Для настройки значения порога вряд ли целесообразно использовать традиционные способы получения обучающих и экзаменационных документальных массивов и требуется специальный подход к формированию выборок.

В данной работе предлагается следующая методика.

1. На сайте ВАК проводится поиск авторефератов докторских диссертаций по заданным специальностям (нами рассматривались диссертации по специальностям 05.13.01—05.13.18 (технические науки) за период с 2008 по 2010 год).

2. На основе списка публикаций в автореферате осуществляется поиск библиографических описаний статей автора в базе данных ИАС "РИНЦ".

3. Статьи, которые имеют в ИАС "РИНЦ" библиографические описания, включаются в выборку для проведения исследований (нами отбиралось от трех до шести статей одного автора (или группы авторов), в первую очередь выбирались публикации, вышедшие в различных изданиях в течение нескольких лет перед защитой докторской диссертации).

При реализации предложенной методики случайным образом было отобрано 24 автореферата и сформирована выборка, содержащая 118 библиографических описаний журнальных статей. Предварительная обработка массивов текстовых данных включала удаление стоп-слов (служебных слов) и представление документов в виде векторной модели (см. формулу (1)).

Формализуем понятие порога:

$$\gamma(X, Y) \leq \alpha.$$

Здесь γ — используемая мера близости между двумя документами X и Y ; α — порог.

В зависимости от γ пороговые значения могут принимать различные значения. Вместе с тем все исследованные меры (расстояние Левенштейна, расстояние Джаро, расстояние Джаро—Винклера, метод шинглов, метод на основе расчета *tf-idf*-взвешивания и косинусоидальной меры, коэффициент Джаккарда) показывают согласованные результаты и выявляют в качестве потенциальных дубликатов одни и те же публикации.

Наиболее легко интерпретируемы и пригодны для проведения сравнительного анализа процедуры, в которых меры близости стремятся к одному предельному значению. Далее рассматриваются расстояние Джаро—Винклера, метод шинглов и коэффициент Джаккарда (см. формулы (3), (4), (6)). Их пороговые значения, отделяющие уникальные публикации от неуникальных, должны приближаться к единице, т. е. $\gamma(X, Y) \rightarrow 1$. В наших исследованиях ввиду наличия в выборке достаточно коротких библиографических описаний (от 30 до 40 терминов) в методе шинглов использовалась последовательность из четырех слов, а масштабный коэффициент, входящий в формулу расчета расстояния Джаро—Винклера, равнялся 0,1 ($p = 0,1$) [17, 18].

Представляется целесообразным в условиях крайне незначительного числа аномальных публикаций осуществлять настройку двух видов порогов: "строгого" и "мягкого". Значение "строгого" порога соответствует случаю "уверенной" идентификации нечеткого дубликата, т. е. можно констатировать практически полное лексическое совпадение библиографических описаний (случай обнаружения дубликата текстового документа в узком смысле). Превышение "мягкого" порога свидетельствует о том, что у документов имеется большое число

общих слов и для определения их статуса (оригинальная публикация или нечеткий дубликат) необходимо провести экспертное оценивание.

Анализ значений мер близости для сформированной выборки позволяет сделать следующие выводы: для большинства публикаций все меры близости принимают значения меньше 0,5. Однако для 10 статей их значения превышают значение 0,5. Проведенная экспертиза этих 10 работ показала, что среди них содержатся две практически полностью совпадающие публикации (отличие заключается в наличии нескольких разных слов в аннотациях, значения всех мер близости превысило 0,9).

Еще для двух статей были зафиксированы достаточно большие значения мер близости ($J = 0,778$; $C(X, Y) = 0,808$; $d_W = 0,923$). Окончательное заключение о статусе этих публикаций можно сделать только после ознакомления с полнотекстовыми версиями и установления степени сходства словосочетаний "детерминированные непрерывно-дискретные системы" — "детерминированные логико-динамические системы" и "устройство управления" — "автомат с памятью". Если в статье рассматриваются различные понятия, то причина высоких значений рассчитанных мер объясняется большим размером аннотаций, которые содержат практически одни и те же формулировки. В случае обнаружения смыслового совпадения указанных словосочетаний можно считать, что идентифицированы дубликаты.

Отметим также, что среди 10 экспертно оцениваемых работ имеются две статьи с различными названиями, но полностью одинаковыми аннотациями ($J = 0,625$; $C(X, Y) = 0,7$; $d_W = 0,824$). Именно вследствие совпадения коротких аннотаций или очень высокой степени "похожести" длинных аннотаций чаще всего получают высокие значения мер близости.

Дадим рекомендации по выбору порогов. Для этого воспользуемся результатами, полученными на сформированной выборке и других массивах текстовых данных (в работах [12, 20] были рассчитаны различные коэффициенты ассоциативности, включая коэффициент Джаккарда, для трех коллекций документов: выборки, состоящей из статей и докладов на конференциях авторов этой работы; выборки из тезисов докладов, сделанных сотрудниками, аспирантами и студентами Института автоматики и вычислительной техники МЭИ на научно-технических семинарах "Современные технологии в задачах управления, автоматики и обработки информации" (Алушта, МАИ, 2006—2009 гг.); выборки, полученной с помощью моделирования коротких текстов).

На наш взгляд, при использовании процедуры выявления нечетких дубликатов можно рекомендовать применять следующие "мягкие" и "строгие" пороги: 0,55 и 0,8 — для коэффициента Джаккарда и метода шинглов; 0,6 и 0,9 — для расстояния Джаро—Винклера.

В целом, результаты проведенных экспериментов позволяют сделать вывод, что даже в небольших массивах журнальных статей не исключено появление аномальных событий — нечетких дубликатов. Показано также, что современные средства обработки и анализа документальной информации способны обеспечить достаточно надежное выявление "подозрительных" публикаций.

Вместе с тем в теории распознавания образов важным условием достоверности полученных зависимостей является возможность их подтверждения на новых коллекциях данных. Поэтому в настоящее время основные усилия авторов направлены на формирование на основе предложенной методики больших выборок журнальных публикаций. В такие выборки должны войти все доступные в ИАС "РИНЦ" статьи соискателей степени доктора технических наук по специальностям 05.13.01—05.13.18 за период с 2008 по 2010 год. При проведении дальнейших исследований несомненный интерес представляет разработка комитетных конструкций, позволяющих комбинировать различные меры близости для выработки более точных решений [12]. Кроме того, важно изучить возможность использования для выявления нечетких дубликатов не только библиографических описаний статей на русском языке, но и английских вариантов названий и аннотаций. В перспективе также необходимо предусмотреть обработку и анализ полнотекстовых публикаций (при создании в ИАС "РИНЦ" полнотекстовой базы данных научной периодики можно будет автономно, с минимальным количеством дополнительных экспертных процедур, проводить мониторинг документального потока журнальных публикаций).

Далее в данной работе анализируются факторы, оказывающие существенное влияние на значение порога. Так, различие в длине (размере) документов снижает значения мер близости. Однако в случае наличия дубликатов (или нечетких дубликатов) размеры документов достаточно близки (на наш взгляд, совпадение размеров может служить индикатором неуникальности статей). Меры близости уменьшаются также вследствие введения сокращений, использования синонимов и близких по значению слов ("разработка—синтез—создание", "метод—процедура—алгоритм—технология", и т. п.), а также орфографических ошибок и различного написания одних и тех же слов. Наличие факторов, снижающих значения мер близости, является дополнительным аргументом в пользу использования при мониторинге публикаций "мягкого" порога.

Проведение морфологического анализа, например стэмминга (выделения корней слов), увеличивает значение мер близости. При стэмминге названия двух статей из сформированной ранее выборки "Автоматизированное проектирование вычислительных сетей крупных проектных организаций" и "Авто-

матризованное проектирование вычислительной сети крупной проектной организации" будут совпадать. Как видно из данного примера, проведение морфологического анализа может снизить чувствительность методов к имеющимся различиям и привести к завышению коэффициентов близости. Поэтому в наших исследованиях стэмминг не применялся (заметим, что указанные выше статьи, несмотря на очень близкие названия, были идентифицированы как оригинальные публикации на основе существенных лексических различий в аннотациях).

Другой способ увеличения значений мер близости заключается в выявлении ключевых слов (связанных словосочетаний) на основе анализа всех доступных в ИАС "РИНЦ" статей автора. Полное совпадение базовых словосочетаний можно рассматривать как еще одно свидетельство в пользу их идентичности.

При разработке программно-алгоритмического обеспечения необходимо принимать во внимание особую роль слов-указателей на разнородность. Например, в рассмотренной выборке имелись статьи с полностью одинаковыми названиями. Однако наличие слов-указателей "Часть 1" и "Часть 2" (так же как и различия в аннотациях) являются маркерами взаимосвязи, а не совпадения публикаций.

Заключение

Споры о методах проведения наукометрического анализа журнальной периодики имеют долгую историю. Сегодня актуальность применения этих методов стремительно возрастает, так как с развитием современных технологий появляются принципиально новые возможности обхода и игнорирования общепринятых в научном сообществе правил и норм. Именно по этой причине разработка и внедрение на практике процедур выявления нечетких дубликатов способна существенным образом расширить инструментарий наукометрии и исключить появление в периодических изданиях статей-клонов.

Как было отмечено выше, наиболее разумно реализовать идентификацию нечетких дубликатов в рамках работ по развитию и совершенствованию ИАС "РИНЦ". Отладка данной процедуры в ходе проведения опытной эксплуатации на больших массивах журнальных публикаций позволит уточнить полученные значения порогов, минимизировать число "ложных" срабатываний, более детально изучить факторы, оказывающие существенное влияние на получаемые результаты.

Включение методов обнаружения дубликатов в наукометрический анализ журнальных публикаций, в конечном итоге, будет способствовать повышению уровня научных статей в российских изданиях и обеспечит более точную оценку "качественной" составляющей публикационной активности ученого при прохождении аттестации, избрании по конкурсу, предоставлении диссертации в Совет.

1. **Еременко Г., Арефьев П.** Сын ошибок трудных. Разработчики проекта РИНЦ предлагают ученым добиваться объективности сообщения // Поиск. 2010. 10 сентября. № 37 (1111).
2. **Медведев Ю.** Фальшивый индекс. Погоня за научным валом искажает суть работы ученого // Российская газета. Федеральный выпуск № 5203 (124) от 9 июня 2010.
3. **Новиков Д. А.** Дискуссия о проблемах оценки научных журналов: вводное слово главного редактора // Управление большими системами. 2009. № 27. С. 5–11.
4. **Научная** электронная библиотека eLIBRARY.RU. URL: <http://elibrary.ru>.
5. **Кузьминов Я.** Испытание разрывом // Поиск. 2010. 19 ноября. № 47 (1121).
6. **Marshall E.** Scientific Misconduct — How Prevalent is Fraud? That's a Million-Dollar Question // Science. 2000. N 290. P. 1662–1663.
7. **Sovacool B. K.** Exploring Scientific Misconduct: Isolated Individuals, Impure Institutions, or an Inevitable Idiom of Modern Science? // Journal of Bioethical Inquiry. 2008. N 5. P. 271–282.
8. **Fanelli D.** How Many Scientists Fabricate and Falsify Research? A Systematic Review and Meta-Analysis of Survey Data // PLoS ONE. 2009. N 4 (5). URL: <http://www.plosone.org>.
9. **Maurer H., Kappe F., Zaka B.** Plagiarism — A Survey // Journal of Universal Computer Science. 2006. N 12 (8). P. 1050–1084.
10. **Dreher H.** Automatic Conceptual Analysis for Plagiarism Detection // Issues in Informing Science and Information Technology. 2007. N 4. P. 601–614.
11. **Chandola V., Banerjee A., Kumar V.** Anomaly Detection: A Survey // ACM Computing Surveys. 2009. September. P. 1–72.
12. **Толчеев В. О.** Анализ проблемы и разработка процедуры выявления нечетких дубликатов научных статей по библиографическим описаниям // Информационные технологии. 2011. № 2. С. 17–21.
13. **Наука** и технологии России. 83,2 миллиона на развитие аналитических возможностей РИНЦ. URL: <http://www.strf.ru>.
14. **Солтон Дж.** Динамические библиотечно-информационные системы. М.: Мир, 1979. 557 с.
15. **Толчеев В. О.** Модели и методы классификации текстовой информации // Информационные технологии. 2004. № 5. С. 6–14.
16. **Дербенев Н. В.** Обзор методов выявления нечетких дубликатов библиографических текстовых документов // Труды 18-й Международной научно-технической конференции "Информационные средства и технологии". М.: Изд-во МЭИ. С. 262–265.
17. **Graham A.** String Search: Technical Report TR-92-gas-01. School of Electronic Engineering Science / University College of North Wales. 1992. October. URL: <http://infoscope.ws/string.search/Stephen-92/index.html>.
18. **Рубцов Д. Н., Баракнин В. Б.** О возможности борьбы с дубликатами при запросах к разнородным библиографическим источникам // Труды 11-й Всероссийской научной конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции". Петрозаводск: Изд-во ПетрГУ, 2009. С. 293–298.
19. **Зеленков Ю. Г., Сегалович И. В.** Сравнительный анализ методов определения нечетких дубликатов для Web-документов // Труды 9-й Всероссийской научной конференции "Электронные библиотеки: перспективные методы и технологии, электронные коллекции". Переславль-Залесский: Изд-во ИПС РАН, 2007. С. 166–174.
20. **Дербенев Н. В., Толчеев В. О.** Сравнительный анализ коэффициентов ассоциативности для выявления нечетких дубликатов текстовых документов // Труды 18-й Международной научно-технической конференции "Информационные средства и технологии". М.: Изд-во МЭИ, 2010. С. 266–270.

Н. И. Юсупова, д-р техн. наук, проф., декан,
Р. С. Валеев, канд. техн. наук, ассистент,
 e-mail: ruslan_valeev@inbox.ru,
 Уфимский государственный
 авиационный технический университет

Рациональное размещение грузов в контейнеры с учетом их физических характеристик с помощью роботизированного комплекса

Рассматривается задача размещения разногабаритных грузов в контейнеры, которая возникает при выполнении различных логистических операций, выполняемых с помощью промышленного робота-погрузчика. Представлена математическая двухкритериальная постановка задачи с учетом плотности размещения и требований к расположению центра тяжести. Для ее решения разработан алгоритм плотного размещения грузов в контейнеры, эволюционный алгоритм поиска допустимого расположения центра тяжести загруженного контейнера с учетом физических характеристик грузов, метод определения центра грузов с помощью роботизированного комплекса.

Ключевые слова: плотное размещение грузов, робот-погрузчик, центр тяжести, силомоментный датчик, эволюционный алгоритм, многоуровневый метод

Введение

В статье рассматривается задача размещения разногабаритных грузов, которую необходимо решать при загрузке транспортных средств, при выполнении операций разгрузки и погрузки в логистических хабах — крупных мультимодальных комплексах (складах). Задача управления загрузкой контейнеров характеризуется непрерывным повышением требований к эффективности решения задачи размещения грузов с учетом их физических характеристик, а также требованием полной автоматизации загрузки на базе роботизированных комплексов. К основным физическим характеристикам грузов относят их габаритные размеры, вес, координаты центра тяжести. Совокупность упакованных грузовых единиц называется транспортным пакетом. Контейнер может содержать множество транспортных пакетов. От эффективности загрузки объема контейнера транспортными пакетами зависит стоимость его доставки, а учет технологических требований к смещению центра тяжести загруженного контейнера позволяет снизить риск его потери во время транспортировки; риск связан

с нарушением устойчивости транспортного средства в процессе его перемещения. Управление загрузкой контейнера на базе роботизированного комплекса позволяет повысить качество и надежность выполнения технологических процессов в условиях жестких требований к эффективному использованию объема контейнера и технологических ограничений по смещению центра тяжести загруженного контейнера.

Поскольку данная процедура является трудоемкой и затратной, предлагается осуществить размещение грузов с учетом расположения центра тяжести. Размещение должно проходить в два этапа: получить плотное размещение грузов в контейнерах (первый этап); смещение центра тяжести уже загруженного контейнера должно быть допустимым (второй этап). При выполнении первого этапа немаловажную роль играет экономия занимаемого грузами пространства.

В связи со сказанным возникает следующая двухкритериальная задача, возникающая при размещении грузов в контейнеры: требуется разместить грузы в контейнеры (паллеты, стеллажи склада, отсеки транспортных средств) таким образом, чтобы плотность размещения была максимальной (первый критерий); смещение центра тяжести загруженного контейнера должно быть в границах допустимой области (второй критерий).

Как правило, геометрически товарные единицы или грузы представляют собой параллелепипеды, которые должны быть достаточно плотно размещены в контейнерах заданных размеров. Подобные задачи относятся к задачам размещения [2, 3, 7, 8] и задачам учета физических характеристик грузов [14]. Для решения задачи плотного размещения грузов в контейнере в работе предлагается алгоритм трехмерного размещения, основанный на декомпозиции исходной задачи и сведении ее к задаче двумерного размещения на базе алгоритма ROMPA [4]. Для определения смещения центра тяжести товарной единицы предлагается методика определения центра тяжести на основе силомоментного датчика, прикрепленного к роботу-погрузчику, а также эволюционный алгоритм (1 + 1)-EA-CG для определения расположения центра тяжести загруженного контейнера на базе известного одноэтапного эволюционного алгоритма (1 + 1)-EA [9].

1. Математическая модель задачи плотного размещения грузов в контейнерах на основе рационального расположения центра тяжести

Задачу размещения грузов в контейнерах с рациональным размещением центра тяжести назовем задачей плотного параллелепипедного размещения с учетом расположения центра тяжести в

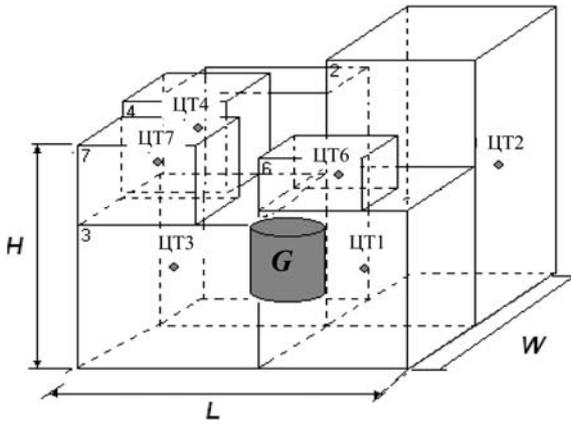


Рис. 1. Задача 3ДВРСГ размещения грузов в контейнере с учетом центра тяжести, где ЦТ1, ..., ЦТ7 — центры тяжести размещенных грузов; G — допустимая область расположения центра тяжести контейнера

границах допустимой области (G), и будем обозначать как 3ДВРСГ (рис. 1).

Формально задача 3ДВРСГ может быть описана с помощью математической постановки задачи 3ДВР [2] и рационального положения центра тяжести. Вводится система координат XYZ , где $(0, 0, 0)$ — начало координат, совпадающее с передней левой нижней вершиной контейнера (параллелепипеда), соответственно (x_i, y_i, z_i) — координаты i -го груза.

Исходная информация представляет собой набор данных $\langle W, L, H, n, w, l, h, m, C \rangle$, $W, L, H, n, w, l, h, m, C \in \mathbb{Z}^+$, где W — ширина контейнера; L — длина контейнера; H — высота контейнера; n — число заданных параллелепипедов (грузов); $w = (w_1, \dots, w_p, \dots, w_n)$ — вектор ширин грузов; $l = (l_1, \dots, l_p, \dots, l_n)$ — вектор длин грузов; $h = (h_1, \dots, h_p, \dots, h_n)$ — вектор высот грузов; $m = (m_1, m_2, \dots, m_n)$ — вектор масс грузов, где m_i — масса i -го груза, $C = (cg_1, cg_2, \dots, cg_n)$ — вектор центров тяжести грузов, где $cg_i = (x_{iЦТ}, y_{iЦТ}, z_{iЦТ})$ — вектор координат центра тяжести i -го груза по осям x, y и z .

Решение задачи 3ДВРСГ представляется в виде набора данных $\langle X, Y, Z, CG \rangle$, $X, Y, Z, CG \in \mathbb{Z}^+$, где $X = (x_1, x_2, \dots, x_p, \dots, x_n)$, $Y = (y_1, y_2, \dots, y_p, \dots, y_n)$, $Z = (z_1, z_2, \dots, z_p, \dots, z_n)$ — векторы координат переднего левого нижнего угла размещенных грузов; $CG = (X_{ЦТ}, Y_{ЦТ}, Z_{ЦТ})$ — координаты расположения центра тяжести загруженного контейнера допустимой области G , т. е. в цилиндре,

$$(x - L/2)^2 + (y - W/2)^2 = R^2, h_1 \leq z \leq h_2, \quad (1)$$

где R — радиус основания цилиндра; h_1, h_2 — границы значения допустимой высоты цилиндра.

Рассмотрим формализованную постановку двух критериев оптимизации задачи 3ДВРСГ.

Первый критерий оптимизации.

Задача 1. Размещение грузов в контейнеры незаданной высоты (на паллеты $H = \infty$).

Дано: Набор исходных данных $\langle W, L, H, n, w, l, h \rangle$.

Требуется: Минимизировать высоту занятой части контейнера

$$H = \max_{i=1, \dots, n} (z_i + h_i) \rightarrow \min_P, \quad (2)$$

где P — множество различных размещений грузов.

При заданных ограничениях:

$$1^{\circ}. ((x_{jl} = x_j) \vee (x_{jl} = x_j + l_j)) \wedge ((y_{jl} = y_j) \vee (y_{jl} = y_j + w_j)) \wedge ((z_{jl} = z_j) \vee (z_{jl} = z_j + h_j)),$$

где (x_{jl}, y_{jl}, z_{jl}) — координаты l -й вершины j -го груза для $j = 1, \dots, n$.

$$2^{\circ}. (x_i \geq (x_j + l_j) \vee x_j \geq (x_i + l_i)) \vee (y_i \geq (y_j + w_j) \vee y_j \geq (y_i + w_i)) \vee (z_i \geq (z_j + h_j) \vee z_j \geq (z_i + h_i))$$

$$3^{\circ}. (x_i \geq 0) \wedge (y_i \geq 0) \wedge (z_i \geq 0) \wedge (L \geq (x_i + l_i)) \wedge (W \geq (y_i + w_i))$$

для $i = 1, \dots, n$.

Задача 2. Размещение грузов в контейнеры ограниченной высоты.

Дано: Набор исходных данных $\langle W, L, H, n, w, l, h \rangle$.

Требуется: Минимизировать число загруженных контейнеров:

$$N \rightarrow \min_P, \quad (3)$$

где P — множество различных размещений грузов.

При заданных ограничениях:

$$1^{\circ} - 3^{\circ} \text{ задачи 1.}$$

$$4^{\circ}. (x_i \geq 0) \wedge (y_i \geq 0) \wedge (z_i \geq 0) \wedge (L \geq (x_i + l_i)) \wedge (W \geq (y_i + w_i)) \wedge (H \geq (z_i + h_i)).$$

Второй критерий оптимизации.

Дано: Набор исходных данных $\langle X, Y, Z, m, C \rangle$, где X, Y, Z — векторы координат переднего левого нижнего угла размещенных грузов.

Требуется: Минимизировать отклонение центра тяжести загруженного контейнера от центра (точки пересечения диагоналей)

$$d = \sqrt{(X_{ЦТ} - L/2)^2 + (Y_{ЦТ} - W/2)^2 + (Z_{ЦТ} - H/2)^2} \rightarrow \min, \quad (4)$$

где

$$\begin{aligned} X_{ЦТ} &= \frac{\sum_{i=1}^n (x_{iЦТ} m_i)}{\sum_{i=1}^n m_i} \\ Y_{ЦТ} &= \frac{\sum_{i=1}^n (y_{iЦТ} m_i)}{\sum_{i=1}^n m_i} \\ Z_{ЦТ} &= \frac{\sum_{i=1}^n (z_{iЦТ} m_i)}{\sum_{i=1}^n m_i} \end{aligned} \quad (5)$$

при следующих ограничениях (нахождения центра тяжести в границах области G):

$$\begin{cases} (x - X_{\text{ЦТ}})^2 + (y - Y_{\text{ЦТ}})^2 \leq R^2, \\ h_1 \leq Z_{\text{ЦТ}} \leq h_2. \end{cases} \quad (6)$$

2. Алгоритм решения задачи плотного размещения грузов в контейнерах

Предлагается метод решения задачи размещения грузов в контейнеры (*3 Dimensional Bin Packing Without Layer, 3DBPWL*), основанный на ее сведении к задаче двухмерного размещения с дополнительными ограничениями, с использованием алгоритма *ROMPA* [4].

Пусть имеется конечный набор грузов с размерами $l_i, w_i, h_i, i = 1, \dots, n$, а также контейнер длины L , ширины W , высоты H . Плотное размещение формируется на основании стратегии, подробно описанной в работе [2]. На этой стратегии основан предлагаемый алгоритм *3DBPWL*.

Алгоритм *3DBPWL*

Шаг 1. Прямоугольник $I \times W$ — основание контейнера — заполняется прямоугольниками $l_i \times w_i$, являющимися основаниями грузов, согласно алгоритму *ROMPA* до тех пор, пока не останется пустых мест для размещения очередного прямоугольника. Если не все грузы упакованы, то идти на шаг 2.

Шаг 2. До тех пор, пока не размещены все оставшиеся грузы, выполнить:

2.1. Найти наибольшие по площади прямоугольники — основания с размерами $L_k = l_{h_{\min}}$, $W_k = w_{h_{\min}}$, являющиеся верхней гранью самого низкого груза из уже размещенных, которые не пересекаются с основаниями уже размещенных грузов, и выполнить их размещение оставшимися грузами согласно алгоритму *ROMPA*. Так, на примере рис. 2 на шаге 1 в прямоугольнике — основании размерности $L \times W$ размещены грузы 1 и 2. Для выполнения шага 2 будет выбрано верхнее основание

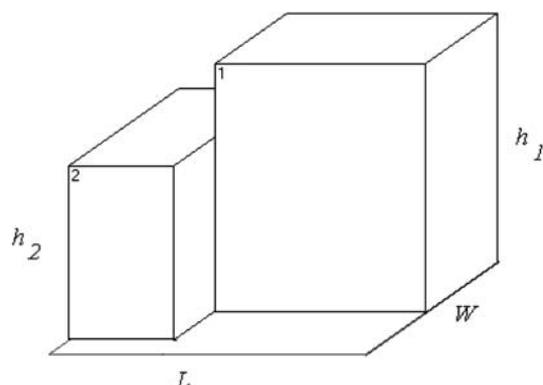


Рис. 2. Размещение грузов в основании контейнера размера LW

груза 2 высоты h_2 для размещения некоторых из оставшихся грузов.

2.2. Вычислить значение целевой функции упаковки (H , если решается задача *3DBP1*, либо N , если решается задача *3DBP2*).

2.3. Если получено размещение с лучшим значением целевой функции, то идти на шаг 3, в противном случае выполнить шаг 1.

Шаг 3. Конец алгоритма.

Было проведено сравнение результатов работы алгоритма *3DBPWL* с результатами алгоритмов, приведенных в работах [2, 11] на тех же тестовых примерах. В эксперименте использовался генератор тестовых примеров, основанный на методике, предложенной Г. Вешером для исследования задач одномерного размещения. Параметры для генерации тестовых примеров были следующие:

- длина основания контейнера $L = 150$;
- ширина основания контейнера $W = 100$;
- число грузов $n = 500$;
- нижнее ограничение длины грузов $v_1 = 0,25$;
- верхнее ограничение длины грузов $v_2 = 0,4; 0,6; 0,8$ ($l_i \geq v_1 L, l_i \leq v_2 L, i = 1, \dots, m$);
- нижнее ограничение ширины грузов $\omega_1 = 0,25$;
- верхнее ограничение ширины грузов $\omega_2 = 0,4; 0,6$ ($w_i \geq \omega_1 L, w_i \leq \omega_2 L, i = 1, \dots, m$);
- нижнее ограничение высоты грузов $\eta_1 = 0,2$;
- верхнее ограничение высоты грузов $\eta_2 = 0,25; 0,6$ ($h_i \geq \eta_1 L, h_i \leq \eta_2 L, i = 1, \dots, m$).

Для каждого из этих классов было отобрано по 10 тестовых примеров:

- мелкие грузы ($v_1 = \omega_1 = 0,1; \eta_1 = 0,2$ и $\eta_2 = 0,25$);
- средние грузы ($v_1 = \omega_1 = 0,2; \eta_1 = 0,2$ и $\eta_2 = 0,25$);
- разнородные грузы ($v_1 = \omega_1 = 0,1; \eta_1 = 0,2$ и $\eta_2 = 0,25$).

Для каждого класса задач было сгенерировано по 10 тестовых примеров. В тестировании принимали участие алгоритмы: *GABD* (классический генетический алгоритм), *EABD* ("наивный" эволюционный алгоритм) [11], *3DBPWL*, МБА (модифицированный бесслынный алгоритм) [2]. Оценка эффективности алгоритмов вычислялась по коэффициенту размещения (КРА). На рис. 3 на диаграмме приведены результаты тестируемых алгоритмов.

По результатам численного эксперимента можно сделать следующие выводы:

- в классах мелкие и разнородные грузы лучшие результаты показали МБА и *3DBPWL* соответственно;
- в классах с мелкими грузами наиболее эффективно были решены задачи алгоритмами МБА, *3DBPWL* и *EABD*;
- в классах со средними и разнородными грузами хорошо проявил себя алгоритм на базе алгоритма динамического перебора МБА и алгоритм *3DBPWL*.

3. Метод нахождения центра тяжести грузовой единицы с использованием силомоментного датчика

Для вычисления центра тяжести нагруженного контейнера необходимо прежде всего знать центр тяжести каждой товарной единицы. Для его определения предлагается метод, основанный на показаниях силомоментного датчика, прикрепленного к роботу-манипулятору. Далее рассматриваются силомоментные датчики [1, 12, 13], измеряющие силу и соответствующий момент силы по трем направлениям (степеням свободы). Они находят применение, в первую очередь, в робототехнике для измерения появляющихся сил при взаимодействии робота с объектом, например, когда робот должен поднять объект (груз) для размещения его в контейнер (рис. 4, а, б).

На рис. 5 приведена роботизированная линия склада по формированию размещения грузов в контейнеры (на паллеты), где 1 — транспортер подачи груза; 2 — рабочий стол; 3 — робот-укладчик; 4 — автоматический магазин поддонов; 5 — транспортер укладки груза на поддон (паллет); 6 — промежуточный паллетный транспортер; 7 — фотобарьер безопасности; 8 — защитное ограждение; 9 — автоматический паллетоупаковщик; 10 — промежуточный выходной паллетный транспортер; 11 — принтер-апликатор.

Рассматриваемый силомоментный датчик JR3-45E12A измеряет силу и соответствующий момент силы по трем направлениям, т. е. имеется набор данных $[F_x, F_y, F_z, M_x, M_y, M_z]$ [5, 6]. Матрица калибровки K используется для вычисления значений этих данных:

$$K = \begin{pmatrix} 0 & 0 & k_{13} & 0 & 0 & 0 & k_{17} & 0 \\ k_{21} & 0 & 0 & 0 & k_{25} & 0 & 0 & 0 \\ 0 & k_{32} & 0 & k_{34} & 0 & k_{36} & 0 & k_{38} \\ 0 & 0 & 0 & k_{44} & 0 & 0 & 0 & k_{48} \\ 0 & k_{52} & 0 & 0 & 0 & k_{56} & 0 & 0 \\ k_{61} & 0 & k_{63} & 0 & k_{65} & 0 & k_{67} & 0 \end{pmatrix}. \quad (7)$$

Элементы k_{ij} матрицы калибровки являются коэффициентами калибровки и, как

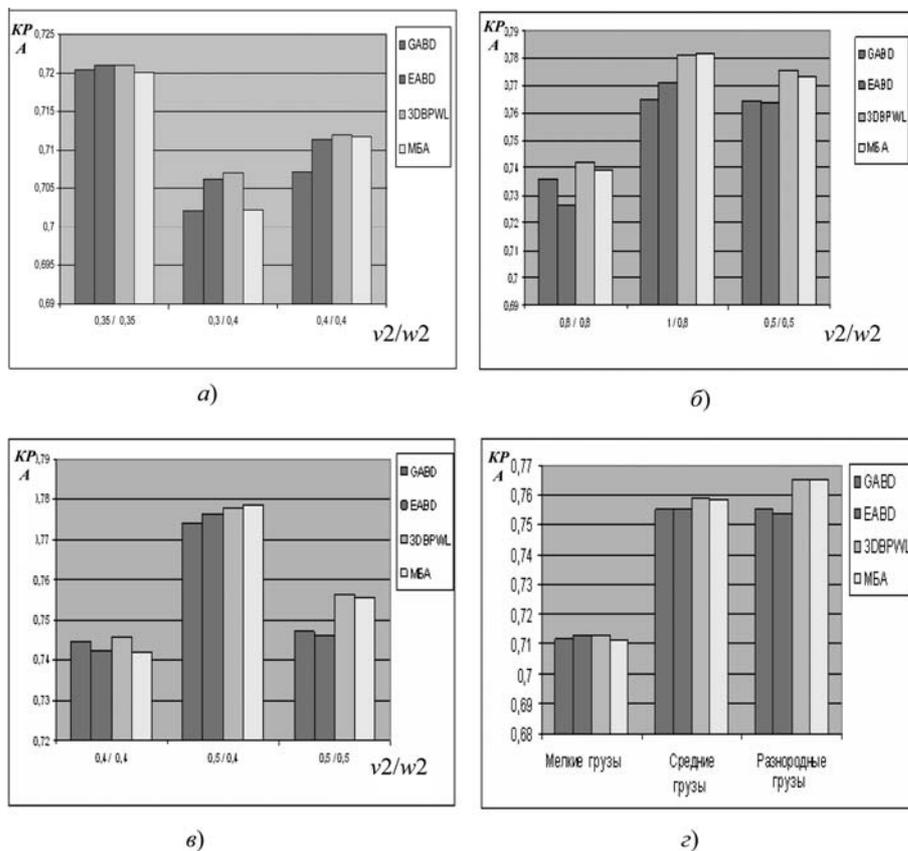


Рис. 3. Результаты работы алгоритмов трехмерного размещения при решении следующих классов задач: а — класс мелких грузов; б — класс разнородных грузов; в — класс средних грузов; з — средние значения работы алгоритмов

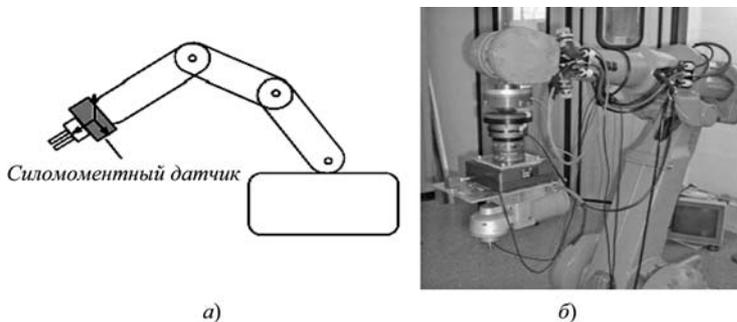


Рис. 4. Использование силомоментного датчика для измерения сил между объектом и роботом (а). Промышленный робот IRB 2400 с прикрепленным к нему силомоментным датчиком JR3 (б)

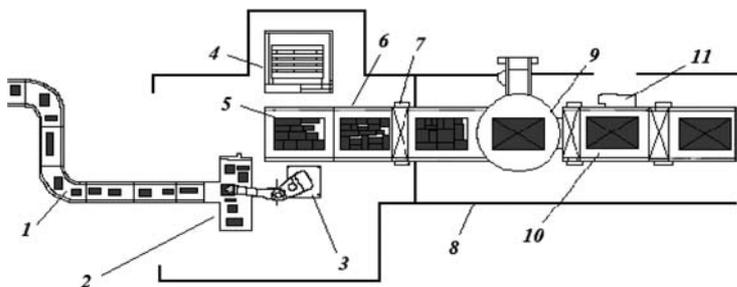


Рис. 5. План участка укладки товарных единиц

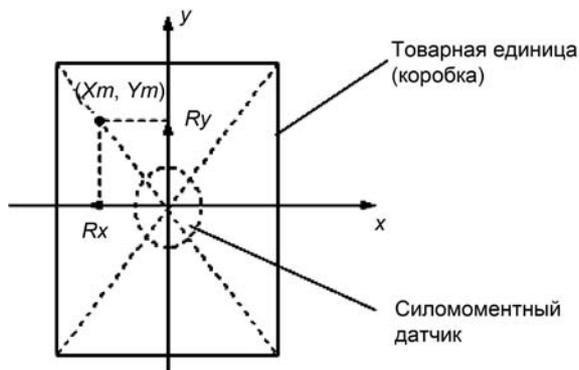


Рис. 6. Определение координат центра тяжести товарной единицы (вид сверху)

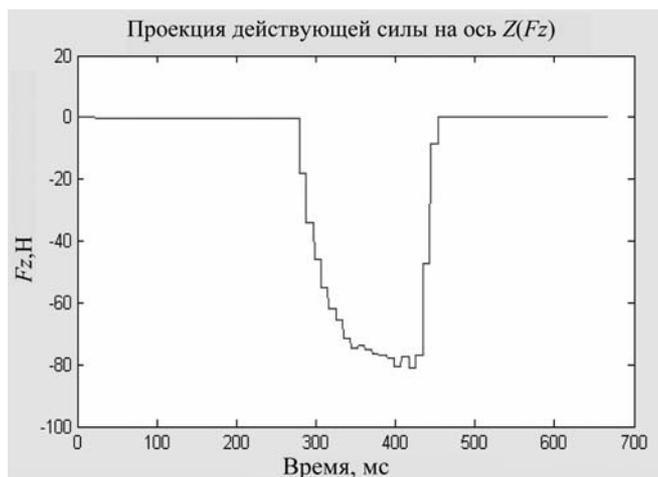
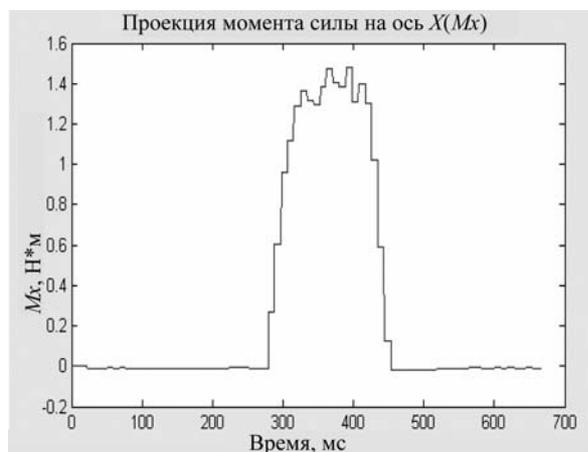


Рис. 7. Изменение действия силы во времени в направлении оси Z

Координаты центров тяжести товарной единицы (коробки)

F_z , Н	-5,45	-11,64	-19,73	-22,19
M_x , Н·м	0	0,63	-0,92	1,07
M_y , Н·м	0	0,71	-0,43	-0,31
$(x_{ЦТ}, y_{ЦТ}, z_{ЦТ})$, см	(7,85; 6,95; 5,3)	(1,76; 1,54; 5,3)	(10,02; 11,61; 5,3)	(9,24; 2,13; 5,3)



правило, определяются производителем силомоментного датчика. Значения $[F_x, F_y, F_z, M_x, M_y, M_z]$ вычисляются из следующего уравнения:

$$\begin{pmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{pmatrix} = K \begin{pmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ W_7 \\ W_8 \end{pmatrix} \quad (8)$$

Значения переменных W_i равны измеренным значениям момента и силы.

Метод вычисления центра тяжести товарной единицы с помощью силомоментного датчика [1, 5, 6] заключается в нахождении координат центра тяжести коробки $(x_{ЦТ}, y_{ЦТ}, z_{ЦТ})$ с помощью значений сил и соответствующих моментов. Координаты $(x_{ЦТ}, y_{ЦТ}, z_{ЦТ})$ соответствуют координатам радиус-векторов плеч сил, которые находятся из выражения

$$|M| = |F| \cdot |R| \cdot |\sin \phi|, \quad (9)$$

где $\phi = 90^\circ$; M — момент силы товарной единицы; R — плечо силы; F — сила, в данном рассмотрении вес товарной единицы.

Таким образом, значения радиус-вектора плеча веса прямоугольной коробки при соответствующей системе координат (рис. 6) определяются из выражения (9), т. е. $|R_x| = |M_y|/F_z$ и соответственно

$$\begin{aligned} |R_x| &= |M_y|/|F_z|, \\ |R_y| &= |M_x|/|F_z|. \end{aligned} \quad (10)$$

Приведем пример расчета центра тяжести товарной единицы следующих размеров: длина $l = 15,7$ см, ширина $w = 13,9$ см, высота $h = 5,3$ см. Координаты несмещенного центра тяжести (x, y, z) соот-

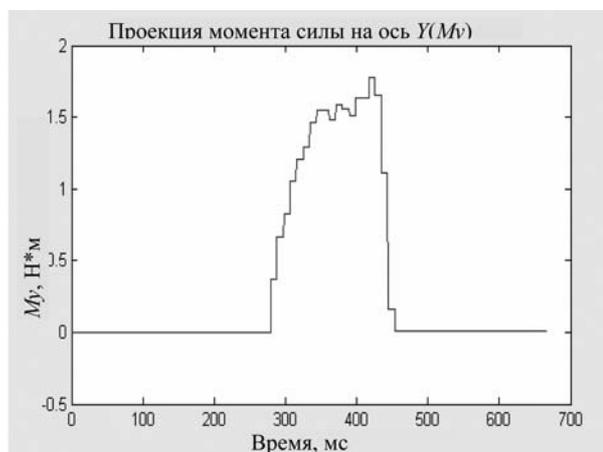


Рис. 8. Изменение действия момента силы во времени в направлении осей X, Y

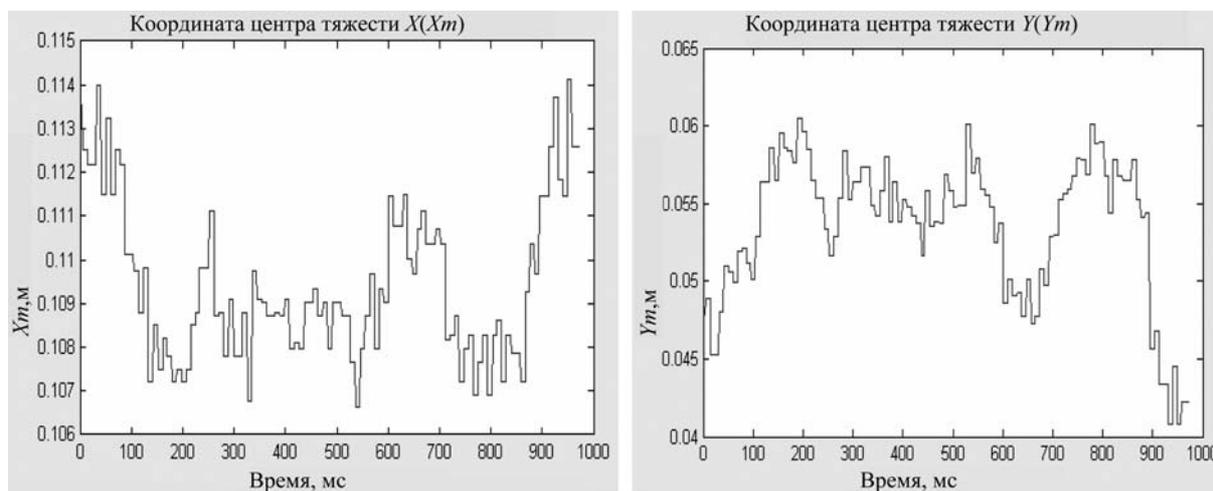


Рис. 9. Изменение координат X , Y центра тяжести коробки во времени

ветственно равны (7,85; 6,95; 2,65) относительно левого нижнего угла коробки. Смещение центра тяжести находим из выражения (10), оно рассматривается в продольном и поперечном направлениях (см. таблицу).

На рис. 7 показано, что на определенном временном интервале [2,8; 4,5] секунд происходит взвешивание груза. Центр тяжести смещен (изначально считается, что он находится на пересечении диагоналей, т. е. груз поднимается в этой точке), так как имеются моменты M_x , M_y соответственно в направлении осей X , Y (рис. 8). На рис. 9 показано изменение координат центра тяжести при его смещении во временном интервале [1; 9] секунд.

Результаты эксперимента показали высокую эффективность алгоритма определения смещения центра тяжести и позволяют сделать вывод, что применение данного датчика для решения задачи определения веса и координат центра тяжести груза, укрепленного на захвате робота-манипулятора, обеспечивает решение задачи управления погрузкой в контейнеры.

4. Размещение грузов в контейнерах на основе эволюционного алгоритма

Предлагается эволюционный алгоритм (1 + 1)-EA-CG, основанный на известном алгоритме (1 + 1)-EA, который на практике при решении ряда задач комбинаторной оптимизации показывает лучшие результаты в классе эволюционных алгоритмов [9]. Перед началом работы алгоритма (1 + 1)-EA-CG должны быть определены центры тяжести каждого из грузов методом, описанным в разд. 3. В качестве начального решения предлагаемого алгоритма (1 + 1)-EA-CG выступает центр тяжести загруженного контейнера d^0 , вычисленного по формуле (5) и решение задачи трехмерного размещения (3DBP1 или 3DBP2) с помощью алгоритма 3DBPWL. Оператор мутации выполняет поворот груза на угол 180° , изменяя положение его центра

тяжести на противоположное. Таким образом, получается новое решение d' . В качестве целевой функции $f(d)$ выступает отклонение центра тяжести загруженного контейнера от границ допустимой области, вычисляемого по формуле (4).

Если оператор мутации на итерации t приблизил значения центра тяжести загруженного контейнера к точке $(W/2; L/2; H/2)$, т. е. целевая функция потомка лучше родительской, тогда на следующей итерации проводится замена родительской особи потомком.

Рассмотрим далее подробнее общую схему данного алгоритма.

Алгоритм (1 + 1)-EA-CG.

1. Генерация начального решения d^0 .
2. Для $t = 0$ до $t_{\max} - 1$ выполнить:
 - 2.1. $d' = \text{Мутация}(d^t)$: поворот заданного числа грузов на угол 180° ;
 - 2.2. Если $f(d') < f(d^t)$, то $d^{t+1} := d'$.
 - 2.3. Если $f(d') = f(d^t)$, $d^{t+1} := d'$ с вероятностью p , иначе $d^{t+1} = d^t$.
 - 2.4. $t := t + 1$.
3. Выбирается загруженный контейнер с наименьшим отклонением положения центра тяжести (лучший результат работы алгоритма).

В статье предлагается многоуровневый метод решения задачи размещения грузов в контейнере с учетом их физических характеристик, основанный на решении двухкритериальной задачи дискретной оптимизации на основе последовательной комбинации решения задачи плотного размещения грузов в контейнере и размещения их с учетом физических характеристик. В отличие от известных методов размещения грузов учитывается критерий плотного размещения грузов в заданном объеме контейнера, а также решается задача переупаковки грузов на основе их перестановок и вращений для удовлетворения критерию расположения центра тяжести контейнера в границах заданной области G .

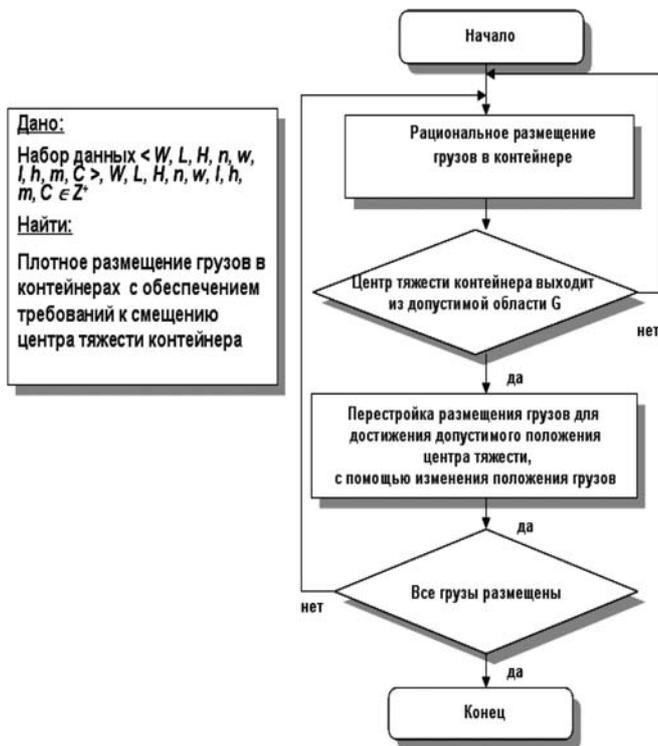


Рис. 10. Многоуровневый метод размещения грузов в контейнере с учетом их физических характеристик

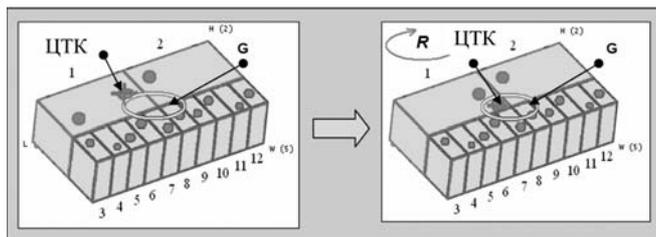


Рис. 11. Пример решения задачи многокритериальной оптимизации на базе разработанного метода погрузки контейнеров (ЦТК — центр тяжести контейнера; R — поворот грузовой единицы; G — допустимая область расположения центра тяжести контейнера)

Обобщенное представление алгоритмической структуры решения задачи размещения грузов в контейнерах с учетом их физических характеристик представлено на рис. 10.

На рис. 11 представлен пример реализации предложенного метода размещения контейнеров на базе разработанных алгоритмов для 12 грузовых единиц.

Заключение

- В заключении можно подвести следующие итоги:
- предложена математическая модель задачи плотного размещения грузов в контейнерах с учетом допустимого положения центра тяжести;
 - разработан и исследован метод решения задачи плотного размещения грузов, основанный на сведении ее к задаче двумерного размещения;
 - предложен метод нахождения центра тяжести грузов с помощью роботизированного комплекса;

- разработан алгоритм учета положения центра тяжести в границах допустимой области загруженного контейнера с использованием эволюционного алгоритма;
- проведен вычислительный эксперимент, позволяющий сравнить и проанализировать эффективность различных алгоритмов;
- приведен многоуровневый метод решения задачи размещения грузов в контейнере с учетом их физических характеристик.

Разработанные алгоритмы могут использоваться для дальнейших исследований в области размещения различных грузов на базе робототехнического комплекса.

Работа поддержана грантами РФФИ 11-07-00579, 09-07-00408, 11-07-00687-а, НШ-65497.2010.9, и хоздоговором ИФ-ВК-03-10 ХК.

Список литературы

1. Юсупова Н. И., Валеев Р. С. Размещение грузов с учетом их физических характеристик с помощью роботизированного комплекса // Компьютерные науки и информационные технологии (CSIT'2010): Матер. 12-й Международ. конф. М.-СПб. 2010.
2. Петунин А. А., Валеева А. Ф. Параллелепипедная упаковка: алгоритмы, численный эксперимент и практические расчеты // Информационные технологии. 2007. № 12. С. 21—29.
3. Жукон Л. А., Корчевская О. В. Метод плоскостей: численный эксперимент для задач двух- и трехмерной ортогональной упаковки // Информационные технологии, 2008. № 11. С. 41—45.
4. Валеев Р. С. Локальный поиск ортогональных упаковок с использованием задачи о (0-1)-рюкзаке // Проблемы оптимизации и экономические приложения: Материалы всерос. конф. Омск, 2009. С. 215.
5. Valeev R., Hein B., Woern H. Rational placement of boxes in the container with the requirements of the position of center of gravity based on robotic complex // Дискретная оптимизация и исследование операций. Материалы Российской конференции. Алтай, 2010. С. 111.
6. Valeev R. S., Hein B., Woern H. Rational placement of cargoes with the use of a force-torque sensor on robotic warehouse line // Proc. of the 12th International Workshop on Computer Science and Information Technologies. Moscow—St. Petersburg, 2010. Vol. 2. P. 16—20.
7. Pisinger D. Heuristics for the container loading problem // European Journal of Operational Research. 2002. № 141. P. 382—392.
8. Moura A., Oliveira J. F. A grasp approach to the container-loading problem // IEEE Computer Society. 2005. July/August. P. 50—57.
9. Борисовский П. А., Еремеев А. В. О сравнении некоторых эволюционных алгоритмов // Автоматика и телемеханика. 2004. № 3. С. 3—9.
10. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. М.: Мир, 1985. 512 с.
11. Мухачева Э. А., Сурначев М. Ю. Задача параллелепипедной упаковки: декодер на базе блочных структур // Принятие решения в условиях неопределенности: Межвуз. науч. сб. Уфа: УГАТУ. 2005. С. 51—55.
12. Garcia J. G., Robertsson A., Ortega J. G., Johansson R. Automatic Calibration Procedure for a Robotic Manipulator Force Observer // International Conference on Robotics and Automation. Barcelona, Spain. 2005. April.
13. Garcia J. G., Robertsson A., Ortega J. G., Johansson R. Force and Acceleration Sensor Fusion for Compliant Robot Motion Control // International Conference on Robotics and Automation. Barcelona, Spain. 2005. April.
14. Романов А. Н., Гиревка Ф. И., Заворотный Д. Г. Определение смещения центра тяжести груза в железнодорожном вагоне // Промышленные измерения. Контроль. Автоматизация. Диагностика (ПиКАД). 2007. № 3. С. 34—35.

А. В. Болховитянов, аспирант,
МГТУ им. Н. Э. Баумана,
e-mail: alex.daiiv@gmail.com,

С. В. Гусев, программист,
НОРСИ-ТРАНС,

А. М. Чеповский, канд. техн. наук, доц.,
НИУ ВШЭ

Модель и база знаний глагольного управления в предложениях на русском языке

Предлагается модель, предназначенная для описания отношений глагола с устойчивыми словосочетаниями при анализе предложений на русском языке. Описывается созданная база знаний для описания синтаксических единиц русского языка. Модель предназначена для выделения объектов и извлечения знаний из текстов на русском языке.

Ключевые слова: обработка текстов на естественном языке, глагольное управление, синтаксический анализ

Введение

Для широкого класса задач информационного поиска в текстах на русском языке и классификации текстов на естественном языке важнейшей задачей стоит разрешение проблемы корректного и полного построения графа синтаксического разбора, распознавание структурных единиц текста. Важнейшим элементом данной проблемы является задача анализа предложений русского языка на основе глагольного управления в целях определения наличия связи между глаголом и словосочетанием.

Теория глагольного управления русского языка является активно развивающейся дисциплиной [1], построенной на иерархии классов и подклассов глаголов по их моделям управления и ориентированной на решения таких сложных задач прикладной лингвистики, как задача машинного перевода текста на естественном языке. Для решения задачи извлечения объектов и знаний об объектах задача синтаксического анализа может быть упрощена по сравнению с решением задачи машинного перевода текста на естественном языке. Эффективным подходом для решения инженерных задач анализа текста является, по нашему мнению, коммуникативная грамматика русского языка [2, 3].

Методики и алгоритмы на основе коммуникативной грамматики русского языка

Коммуникативная грамматика русского языка рассматривает обобщенное значение слова, опре-

деляющее синтаксические возможности слова и их связь с семантикой. В работе [3] приводятся различные способы образования глагольных сочетаний. Каждый такой вариант образуется из определенного глагола и описания употребления словосочетания, которое с этим глаголом может быть связано. Такие варианты формирования сочетаний называют ограничениями. Описание ограничения состоит из предлога и падежа зависимого словосочетания. Каждому глаголу может быть сопоставлено как одно, так и, в общем случае, несколько ограничений, в совокупности образующих модель глагольного управления. Вводится понятие синтаксемы — минимальной семантико-синтаксической единицы русского языка, выступающей одновременно как носитель элементарного смысла и как конструктивный компонент более сложных синтаксических построений, характеризуемой, следовательно, определенным набором синтаксических функций. Смысловая связь между понятиями предложения может быть описана глаголами.

Диапазон синтаксических возможностей каждой синтаксемы определяется ее функциональными свойствами, принадлежностью к тому или иному функциональному типу. Установление семантико-синтаксических связей позволяет сформировать схему ситуации, описываемой во фразе, показать устройство языка.

Взаимодействия морфологических, лексических и синтаксических единиц связываются функциональным синтаксисом и функциональной морфологией [4]. Функциональное описание является эффективным не только с позиций теоретической лингвистики [4], но и для информационного описания в задачах компьютерной лингвистики [5]. Последовательное применение функциональной модели позволяет сформулировать алгоритмы синтаксического анализа текста на естественном языке [6, 7].

Для всех глаголов, найденных в предложении, ищутся возможные зависимые синтаксемы — либо сочетания "предлог + именная группа", либо именные группы без предлога. Затем выполняется поиск описания зависимой синтаксемы в базе моделей управления, при этом падеж именной группы в предложении должен совпадать с падежом в описании синтаксемы. Если описание требует наличия предлога, то он должен совпадать с предлогом, найденным в предложении. Если модель управления не включает в себя предлог, то зависимая синтаксема должна быть именной группой без предлога. В результат включаются только те зависимые синтаксемы, которым были успешно сопоставлены описания, найденные в базе моделей управления. База моделей управления строится на основе модели синтаксемы.

Функциональная модель синтаксемы

Синтаксема служит конструктивно-смысловым элементом предложения, и ее тип (свободная, связанная, обусловленная) определяет диапазон возможностей употребления данной синтаксемы в предложении.

Глагольное управление — это отображение, ставящее в соответствие каждому глаголу список ограничений:

$$VG: v \rightarrow (r_1, \dots, r_n).$$

Ограничение — это кортеж $r = (\text{preposition}, \text{gc}, \text{type})$, где *preposition* — это предлог, с которым глагол употребляется в предложении; *gc* — падеж завысимог словосочетания; *type* — тип зависимой синтаксемы (свободная, связанная или обусловленная).

В качестве зависимого словосочетания могут выступать именные и предложные группы.

В соответствии с работой [8] под базами знаний понимают совокупность фактов и правил вывода, допускающих логический вывод и осмысленную обработку информации. Как можно видеть из приведенного ранее определения, ограничения являются аналогами правил в базах знаний, так как используются для выделения групп глагольного управления, при этом глаголы и словосочетания выступают здесь в качестве фактов. Поэтому можно говорить о базе знаний глагольного управления русского языка.

Программная реализация

Программная реализация базы моделей глагольного управления подразумевает два основных режима использования. Первый режим — это наполнение базы моделями конкретных глаголов и ее корректирование. Второй режим — это использование наполненной базы в задачах анализа текста. Так как требования, предъявляемые к работе программно-го обеспечения, в этих режимах существенно различаются, программная реализация была разделена на две независимые части, каждая из которых предназначена для работы в своем режиме. Таким образом, программное обеспечение для работы с базой моделей глагольного управления состоит из следующих компонентов:

- графическое пользовательское приложение — рабочее место лингвиста для ручного заполнения базы;
- база данных в собственном формате для использования с графическим приложением;
- библиотека для поиска модели по заданной начальной форме глагола;
- словарь с данными в бинарном формате для библиотеки.

Рабочее место лингвиста является графическим пользовательским приложением и предназначено для ручного наполнения и редактирования базы

моделей лингвистом. Оно предоставляет возможности по добавлению новых глаголов и дополнению и изменению наборов ограничений в их моделях. Вся информация о моделях управления сохраняется в базе данных в собственном бинарном формате, которая хранит информацию в виде, наиболее удобном для загрузки и сохранения изменений, а также хранит дополнительную информацию, необходимую для удобства редактирования базы моделей, но не нужную для поиска.

Для использования данных библиотекой они экспортируются в бинарный словарь, который оптимизирован для операции поиска моделей глаголов при анализе текста и не содержит никаких дополнительных данных. Экспорт данных из базы в словарь осуществляется с помощью рабочего места лингвиста.

Для использования моделей глагольного управления при анализе текста реализована библиотека, предоставляющая программный интерфейс для работы с базой моделей. Во время инициализации библиотеки данные из файла загружаются в память. После этого становится возможным поиск моделей глаголов, присутствующих в словаре, по их начальным формам. Результатом поиска является либо полное описание модели управления соответствующего глагола, либо информация об отсутствии искомого глагола в словаре.

Наполнение базы знаний глагольного управления

Ключевой проблемой анализа синтаксических связей является задача наполнения конкретными шаблонами базы знаний глагольного управления. Печатный словарь [3] содержит данные только о 700 глаголах и связанных с ними моделями глагольного управления, что явно недостаточно для анализа текстов. Поэтому была сделана попытка разработать алгоритм построения моделей управления, позволяющий автоматически дополнять словарь глаголами и их моделями управления.

Попытка автоматизации наполнения базы данных была основана на разборе корпусов текстов, в процессе которого осуществлялся поиск глагола и его именных групп в предложении. Алгоритм использует результаты морфологического разбора текста. Для всех глаголов, найденных в предложении, ищутся предлог (среди пяти соседних слов) и именная группа (среди пяти соседних слов с глаголом). Найденные совокупности "глагол + предлог + именная группа" заполняют базу данных соответствующими записями. Реально такая методика приводит только к созданию базы данных глаголов с конкретными ограничениями и с содержащимися в анализируемых корпусах текстов именными группами. Полнота такого пополнения по результатам наших экспериментов недостаточна.

Кроме этого, возникают следующие проблемы. Определение типа зависимой синтаксемы в общем

случае недостижимо на разумных выборках. Одной из проблем, возникших при реализации алгоритма автоматизированного заполнения, стала омонимия глаголов, для устранения которой приходится ограничивать выбор искусственными ограничениями. Во избежание нахождения избыточных связей, для исключения добавления неправильных связей приходится переходить к вероятностной модели: глагол и его модель управления должны встретиться в тексте заданное число раз, чтобы быть добавленными в результирующую базу данных.

Мы вынуждены констатировать несостоятельность декларированных в работе [7] автоматических методик наполнения базы знаний глагольного управления. Поэтому наполнения словаря опиралось на неавтоматические, "ручные" процедуры, опирающиеся на разработки [3, 9].

Для формирования словника базы знаний глаголов были использованы два источника: синтаксический словарь Г. А. Золотовой [3] и частотные словари [9], построенные по материалам Национального корпуса русского языка [10]. Мы взяли объединение с исключением повторов множества глаголов, описанных в синтаксическом словаре

Г. А. Золотовой [3] мощностью около 700 и множества первых 2000 наиболее частотных глаголов русского языка по данным авторов частотных словарей русского языка [9], что позволяет учесть все слова с числом употреблений на 1 000 000 слов корпуса больше 10. Получился словник, состоящий из 2316 глаголов.

В соответствии с полученным словником заполнялась база знаний. Приведем пример набора ограничений для конкретных глаголов. Модели управления для глагола "поселиться" приведены в табл. 1. Для данного глагола в базе данных указываются такие темы, как "населенные пункты", "персоны". Модели управления для глагола "застраховать" приведены в табл. 2. Для этого глагола характерны темы "недвижимость", "автомобили", "персоны".

Заключение

В данной статье представлены функциональная модель синтаксем и созданная база знаний, описывающая структурные взаимодействия глагола с его синтаксическим окружением в предложениях на русском языке. Данные разработки предназначены для решения прикладных задач обработки текстов. Среди такого рода задач стоит выделить две:

- выделение объектов в текстах на русском языке;
- извлечение знаний из текстов на естественном языке.

Список литературы

1. **Апресян Ю. Д., Богуславский И. М., Иомдин Л. Л., Санников В. З.** Теоретические проблемы русского синтаксиса: Взаимодействие грамматики и словаря. М.: Языки славянской культуры, 2010. 408 с.
2. **Осипов Г. С.** Лекции по искусственному интеллекту. М.: КРАСАНД, 2009. 272 с.
3. **Золотова Г. А.** Синтаксический словарь: Репертуар элементарных единиц русского синтаксиса. М.: Едиториал УРСС, 2006. 440 с.
4. **Мухин А. М.** Функциональный синтаксис, функциональная лексикология, функциональная морфология. СПб.: Наука, 2007. 198 с.
5. **Батура Т. В., Мурзин Ф. А.** Машинно-ориентированные логические методы отображения семантики текста на естественном языке. Новосибирск: Изд. НГТУ, 2008. 248 с.
6. **Чеповский А. М.** Элементы функциональной лингвистики в проблемах поиска и классификации // Перспективы систем информатики. Седьмая Международная конференция памяти академика А. П. Ершова. Рабочий семинар "Наукоемкое программное обеспечение". Новосибирск: ООО "Сибирское научное издательство", 2009. С. 273—274.
7. **Гусев С. В., Шуголь А. А.** Использование базы моделей управления глаголов русского языка при синтаксическом анализе // Технологии Microsoft в теории и практике программирования. Труды V Всероссийской конференции студентов, аспирантов и молодых ученых. Центральный регион. М.: Вузовская книга, 2008. С. 217—218.
8. **Гаврилова Т. А., Хорошевский В. Ф.** Базы знаний интеллектуальных систем. СПб.: Питер, 2000. 384 с.
9. **Ляшевская О. Н., Шаров С. А.** Частотный словарь современного русского языка (на материалах Национального корпуса русского языка). М.: Азбуковник, 2009. 2008 (дата обращения: 25.04.2011). URL: <http://dict.ruslang.ru/freq.php>.
10. **Национальный корпус русского языка.** URL: <http://ruscorp.org.ru/> (дата обращения: 25.04.2011).

Таблица 1

Модели управления для глагола "поселиться"

Предлог	Падеж словосочетания	Тип синтаксемы	Примеры шаблонов конструкций
у	Родительный	Обусловленная	у кого-либо, у чего-либо
в	Предложный	Обусловленная	в ком-либо, в чем-либо
с	Творительный	Свободная	с кем-либо, с чем-либо
над	Творительный	Свободная	над кем-либо, над чем-либо
под	Творительный	Свободная	под кем-либо, под чем-либо
за	Творительный	Свободная	за кем-либо, за чем-либо
между	Творительный	Свободная	между кем-либо, между чем-либо
среди	Родительный	Свободная	среди кого-либо, среди чего-либо

Таблица 2

Модели управления для глагола "застраховать"

Предлог	Падеж словосочетания	Тип синтаксемы	Примеры шаблонов конструкций
от	Родительный	Свободная	от кого-либо, от чего-либо
<без предлога>	Винительный	Свободная	кого-либо, что-либо
на	Винительный	Обусловленная	на кого-либо, на что-либо

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И ДИАГНОСТИЧЕСКИЕ СИСТЕМЫ

УДК 621.771

Л. А. Кузнецов, д-р техн. наук, проф., зав. каф.,
Липецкий государственный
технический университет,
e-mail: kuznetsov@stu.lipetsk.ru

Меры оценки состояния объектов и процессов при мониторинге

Разработаны меры оценки состояния объектов и процессов, мониторинг которых обеспечивается наблюдением множества случайных величин. При мониторинге измеряются значения множества физических величин различной природы. Разработанная методология обеспечивает структуризацию контролируемых величин в виде вектора мониторинга, на основании реализаций которого формируются характеристики оценки состояний и эволюции объектов и процессов. Для состояний и эволюции объекта, представляемого множеством случайных контролируемых величин, вводятся скалярные характеристики, значения которых определяются по ретроспективной информации мониторинга. Характеристики позволяют оценивать актуальное состояние объекта и тенденции его изменения и могут быть использованы при синтезе автоматизированных систем мониторинга объектов и процессов.

Ключевые слова: мониторинг состояния, случайный вектор, ковариационная матрица, энтропия

Введение

Мониторинг различных объектов, состояния оборудования, производственных процессов, технологий, качества продукции и т. п. сводится к контролю значений совокупности величин, отражающих изменение состояния объекта мониторинга. Во многих практических задачах мониторинга контролируемые величины являются случайными, а их количество может быть значительным. Измеренные в процессе мониторинга значения величин позволяют судить о тенденциях эволюции состояния объекта.

В зависимости от содержательного существа задач целями мониторинга может быть контроль нахождения объекта или процесса в заданных границах, сравнение состояний различных подобных объектов, сравнение состояний в различные периоды функционирования и т. п. Мониторинг предполагает систематический контроль значений всех величин, необходимых для оценки состояния наблюдаемого

процесса или объекта. Для формирования выводов о состоянии объекта необходимо определенным образом обработать множество измеренных значений множества контролируемых случайных величин.

Отслеживание эволюции одной случайной величины трудностей не представляет, но когда их число велико и характер изменения отдельных компонент случаен, то представление обобщенной картины эволюции может представлять определенные трудности. Получение суждения о состоянии и тенденциях эволюции процесса в этом случае должно базироваться на характеристиках, отражающих обобщенное (усредненное) представление информации, содержащейся в массивах измеренных значений контролируемых величин.

Мониторинг может осуществляться путем контроля различных физических величин, имеющих существенно различающиеся области значений, поэтому возникает необходимость структуризации их представления и свертки статистической информации, содержащейся в массивах измерений, в некоторые статистики, позволяющие оценить состояние и тенденции изменения контролируемого процесса. Ниже предлагаются статистики и методика их формирования, отражающие обобщенные тенденции изменения многомерных случайных характеристик объекта.

1. Формальное представление контролируемых величин

Пусть состояние представляющего интерес объекта характеризуется величинами u_i , $i = 1, 2, \dots, n$, где n — количество величин. Вследствие того, что это измеряемые величины, они являются метрическими физическими величинами. Для систематизации представления множество величин, контролируемых в процессе мониторинга, целесообразно преобразовать к безразмерному, однообразному виду [1]. Для каждой величины u_i может быть выбрано не-

которое опорное значение u_i^0 , из ограниченной окрестности которого она принимает свои значения при функционировании объекта. Использование этих значений в качестве масштабных единиц обеспечит сопоставимость и однородность исходно различных контролируемых физических величин:

$$\zeta_i = u_i / u_i^0, \quad i = 1, 2, \dots, n, \quad (1)$$

где ζ_i — безразмерное представление величины u_i ; u_i^0 — масштабная единица величины u_i в системе мониторинга.

Безразмерные и равноправные величины ζ_i могут быть объединены в вектор $\zeta = (\zeta_1, \zeta_2, \dots, \zeta_n)$ и может быть введено векторное метрическое пространство мониторинга U , координатные оси которого ассоциируются с величинами u_i , посредством которых осуществляется мониторинг.

Измеренные значения величины u_i обозначим τ_{qi} , где $q = 1, 2, \dots, Q$ — номер измерения (реализации); Q — число реализаций. Измеренные значения в относительном масштабе будут определяться в соответствии с формулой (1):

$$\xi_{qi} = \tau_{qi}/u_i^0. \quad (2)$$

Измеренные значения идентифицируются либо временной, либо пространственной координатой. Совокупность измеренных значений всех контролируемых величин в относительном масштабе изобразится в пространстве мониторинга U точкой и соответствующим ей вектором

$$\xi_q = (\xi_{q1}, \xi_{q2}, \dots, \xi_{qn}) = \xi_{q1}e_1 + \xi_{q2}e_2 + \dots + \xi_{qn}e_n, \quad (3)$$

где e_i — базисные векторы, направленные по контролируемым величинам u_i ; ξ_{qi} — измеренные значения компонент мониторинга в относительном масштабе.

Можно использовать отклонения текущих значений от опорных значений u_i^0 , т. е. величины

$$\Delta\xi_{qi} = (\tau_{qi} - u_i^0)/u_i^0. \quad (2a)$$

Все эти представления для случая $n = 2$, $u = (u_1, u_2)$, иллюстрирует рис. 1.

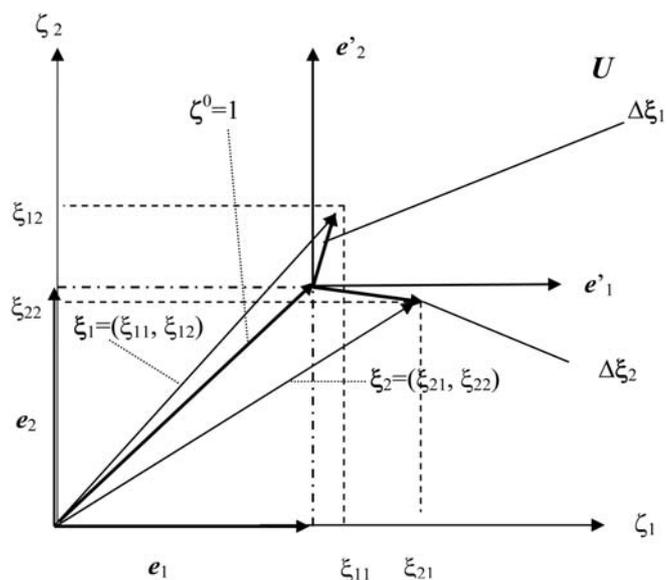


Рис. 1. Геометрическое представление двухмерного пространства мониторинга

В реальных задачах мониторинга число величин, характеризующих состояние объекта, может составлять десятки и сотни.

Использование векторного представления позволяет свернуть информацию и получить скалярные характеристики для отражения тенденций изменения многомерного случайного процесса в функции координаты q .

2. Характеристики состояния и тенденций изменения

Случайный характер контролируемых величин предписывает использование статистик оценки "среднего состояния" и "нестабильности состояния" объекта мониторинга. Статистики получаются по множествам реализаций. Для одномерной величины порция, выборка, проба информации представляется в виде кластера измерений

$$L(l) = \{\xi_q, q = 1, 2, \dots, Q(l)\}, \quad (4)$$

где $Q(l)$ — число элементов в кластере l .

В случае многомерной величины картина изменяется. Выборка имеет вид:

$$L(l) = \{\xi_q, \xi_q = (\xi_{q1}, \xi_{q2}, \dots, \xi_{qn}), q = 1, 2, \dots, Q(l)\}. \quad (4a)$$

Рассмотрение всех величин по отдельности $\mu_i(l)$ и $\sigma_i(l)$ не дает общего представления об изменении состояния объекта и может привести к потере информации. Компоненты вектора мониторинга u могут оказаться взаимосвязанными, что должно отражаться при оценке изменения состояния объекта. При решении многих задач, для поддержки которых и организуется мониторинг, необходимо иметь обобщенные показатели состояния многомерного объекта мониторинга.

В качестве характеристики среднего состояния объекта может быть использована статистика в виде евклидовой нормы средних значений отдельных компонент вектора мониторинга:

$$\| \mu(l) \| = \{ \mu_1^2(l) + \mu_2^2(l) + \dots + \mu_n^2(l) \}^{1/2}, \quad (5)$$

где $\mu_i(l)$ — средние значения отдельных величин, вычисленные по значениям кластера (4a).

Дополнительной показательной характеристикой может быть значение уклонения среднего (5) от некоторого заданного значения, например от опорной точки, показанной на рис. 1 вектором $\zeta^0 = 1$.

Характеристикой разброса для случайного вектора u является ковариационная матрица $R_u(l)$, которая по значениям компонент вектора мониторинга на кластере $L(l)$ определяется известным образом.

Статистики $\| \mu(l) \|$, $R_u(l)$ подобны среднему значению и дисперсии, используемым для характеристики скалярной случайной величины. Однако непосредственное сопоставление ковариационных матриц $R_u(l)$, вычисленных по разным кластерам измерений (для разных l) при значительных n

представляется затруднительным. Для свертки информации, содержащейся в $\mathbf{R}_u(l)$ целесообразно использовать энтропию [2, 3]. Энтропия случайного вектора [4], имеющего гауссово совместное распределение вероятностей, определяется через его ковариационную матрицу следующим образом:

$$\mathbf{H}_u(l) = -\ln \sqrt{\det \mathbf{R}_u(l)} = -\frac{1}{2} \ln \det \mathbf{R}_u(l), \quad (6)$$

где $\det \mathbf{R}_u(l)$ — определитель ковариационной матрицы $\mathbf{R}_u(l)$.

Несмотря на некоторую абстрактность, энтропия имеет достаточно простой содержательный смысл: для нормально распределенного случайного вектора \mathbf{u} величина, равная $\sqrt{\det \mathbf{R}_u}$, пропорциональна объему доверительной области в пространстве \mathbf{u} [5]. Так что характеристика (6) для многомерной случайной величины аналогична среднему квадратическому отклонению для скалярной величины.

Пара величин $[\mu(l), \mathbf{H}_u(l)]$ может характеризовать множество реализаций мониторинга, составляющих кластер $\mathbf{L}(l)$. При этом норма (5) является оценкой среднего значения, а энтропия (6) — мерой разброса измеренных значений величин, составляющих кластер $\mathbf{L}(l)$. Энтропия (6) является монотонной непрерывной функцией, позволяющей сопоставлять нестабильность объекта по кластерам $\mathbf{L}(l)$ измеренных значений контролируемых величин.

3. Градуировка характеристик

Среднее и энтропия являются абстрактными величинами. Для оценки состояния конкретного объекта необходимо осуществить их градуировку. Градуировка (тарирование) может быть выполнена по известным значениям соответствующих величин.

Например, опорное значение $\mathbf{u}^0 = (u_1^0, u_2^0, \dots, u_n^0)$ контролируемых величин (см. рис. 1) может быть принято за единицу отсчета. Для энтропии (6) известное заданное значение может быть получено на основании опорной точки \mathbf{u}^0 следующим образом. При использовании гипотезы о распределении случайных величин u_i по закону Гаусса, средние значения и средние квадратические отклонения полностью определяют распределение. Поэтому опорное распределение вероятностей каждой величины может быть задано в виде:

$$u_i = (N, \mu_i^0 = u_i^0, s_i^0 = \beta_i u_i^0), \quad (7)$$

где μ_i^0 — среднее значение; s_i^0 — среднее квадратическое отклонение; β_i — параметр, посредством которого задается среднее квадратическое отклонение.

При условии взаимной независимости контролируемых величин ковариационная матрица, соответствующая совокупности случайных векторов вида (7), будет диагональной. Ее диагональные элементы равны дисперсиям контролируемых величин, которые при распределении величин по закону Гаусса (7) будут равны $s_i^0 s_i^0$. Опорная ковариационная матрица может быть принята инвариантной по отношению к кластерам в виде:

$$\mathbf{R}_u^0 = \begin{bmatrix} s_1^0 s_1^0 & 0 & \dots & 0 \\ 0 & s_2^0 s_2^0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & s_n^0 s_n^0 \end{bmatrix}. \quad (8)$$

Значение энтропии (6), вычисляемое по этой матрице, будет

$$\begin{aligned} H_u^0 &= -\frac{1}{2} \ln \prod_{i=1}^n s_i^0 s_i^0 = -\frac{1}{2} \sum_{i=1}^n \ln s_i^0 s_i^0 = \\ &= -\sum_{i=1}^n \ln s_i^0 = -\sum_{i=1}^n (\ln \beta_i + \ln u_i^0). \end{aligned} \quad (9)$$

Известные, критические, допустимые и т. п. значения энтропии в соответствии с (9) могут быть заданы через параметры β_i , $i = 1, 2, \dots, n$, и значение опорной точки $\mathbf{u}^0 = (u_1^0, u_2^0, \dots, u_n^0)$. В качестве характеристики уровня нестабильности в определенный период контроля объекта мониторинга может быть использована величина

$$V_u(l) = H_u(l) / H_u^0. \quad (10)$$

Видно, что эта величина выражает текущую энтропию в единицах опорной (заданной и определенной) энтропии, поэтому выражение (10) является характеристикой обобщенной нестабильности объекта мониторинга в единицах опорной нестабильности.

4. Пример

Для примера требуется множество реализаций случайного вектора, в которых присутствует временной дрейф. Используется массив измеренных при производстве тонколистовой стали значений технологических факторов. Массив содержит около 500 записей ($Q_\tau = 500$) реальных значений технологических факторов и показателей качества. Факторы трактуются случайными взаимно независимыми величинами, поэтому их контроль представляет пример мониторинга случайного объекта — металлургической технологии. Для иллюстрации из общего числа (более 30) отобраны 10 технологиче-

ских факторов. В данные для явной демонстрации и наглядности временного дрейфа вводилась аддитивная случайная составляющая в виде случайной нормально распределенной величины с изменяемыми параметрами

$$\omega_i(k) = N\{\alpha k \mu_i, (\beta k \sigma_i)\}, \quad (11)$$

где $\alpha \ll 1$, $\beta \ll 1$ — параметры, управляющие величиной шума среднего значения и дисперсии соответственно; μ_i , σ_i — оценки среднего и среднего квадратического отклонения, полученные на данных реального массива; $k = 1, 2, \dots$, — порядковый номер записи в массиве данных Q_τ .

Зашумленные значения технологических факторов τ_i^{III} формировались добавлением шума (11) к измеренным значениям технологических факторов из исходного массива:

$$\tau_i^{III}(k) = \tau_i(k) + \omega_i(k), \quad (12)$$

где $\tau_i(k)$ — значения из эмпирического массива данных.

Массив разделялся на кластеры $L(l)$, $l = 1, 2, \dots, 16$, (по 30 реализаций в каждом), для которых вычислялись величины $[\mu(T), \mathbf{H}_\mu(T)]$, характеризующие дрейф. Для наглядной иллюстрации были смоделированы три сценария развития дрейфа контролируемого векторного процесса. Первый вариант соответствует дрейфу средних значений компонент контролируемого векторного процесса. Он эмитируется внесением в соответствии с (11) и (12) в исходный массив наблюдений аддитивного шума, формируемого при значениях параметров $\alpha = 10^{-6}$, $\beta = 0$. Из выражения (11) видно, что параметр α управляет значением математического ожидания шума, а параметр β , управляет значением дисперсии шума. Поэтому первый сценарий соответствует изменению во времени средних значений компонент контролируемого процесса.

Второй сценарий реализуется добавлением шума (11) с параметрами $\alpha = 0$, $\beta = 0,0005$. Он соответствует нарастанию колебания компонент процесса относительно средних значений, т. е. нарастанию нестабильности процесса. Третий сценарий эмитирует вариант, являющийся суперпозицией двух первых, когда с течением времени изменяются и средние значения отдельных процессов и их нестабильность относительно средних значений.

Он реализуется добавлением шума (12) с параметрами в (11) $\alpha = 10^{-6}$, $\beta = 0,0005$.

Для оценки нестабильности векторного случайного процесса может использоваться ковариационная матрица $\mathbf{R}_u(l)$. Для наглядности в таблице представлен фрагмент ковариационной матрицы, рассчитанной для кластера 10 для пяти элементов случайного вектора.

Для всех 10 компонент случайного вектора, используемых в примере, размеры матрицы возрастут соответственно. На практике размерность вектора, мониторинг которого осуществляется, может быть существенно больше. Сопоставлять нестабильность непосредственно по матрицам даже пятого порядка достаточно трудно.

Можно предположить, что проблема оценки дрейфа случайного процесса может быть решена использованием графического представления. На рис. 2 для иллюстрации представлены результаты расчета средних значений 10 величин для всех 16 кластеров. При этом шум вычислялся по сценарию 3. Результаты расчета на рис. 2 (и на последующих рис. 3, 4) представлены в виде условных графиков. Условность графиков в том, что в действительности имеются значения соответствующих величин только в дискретных точках, а между точками они не определены.

Рис. 2 показывает, что из него достаточно сложно сделать выводы об общей эволюции контролируемого 10-мерного процесса во времени. При боль-

Фрагмент ковариационной матрицы для кластера 10

	Заправочная скорость	Скорость прокатки	t° конца прокатки	t° смотки	Концентрация марганца
Заправочная скорость	-0,0002	-0,0023	-0,0001	-0,0009	0,0020
Скорость прокатки	-0,0116	-0,0225	0,0010	0,0029	0,0019
t° конца прокатки	0,0025	0,0052	-0,0002	-0,0012	0,0006
t° смотки	0,0015	-0,0070	0,0001	-0,0008	0,0043
Концентрация марганца	0,0243	0,0181	-0,0007	0,0015	-0,0002

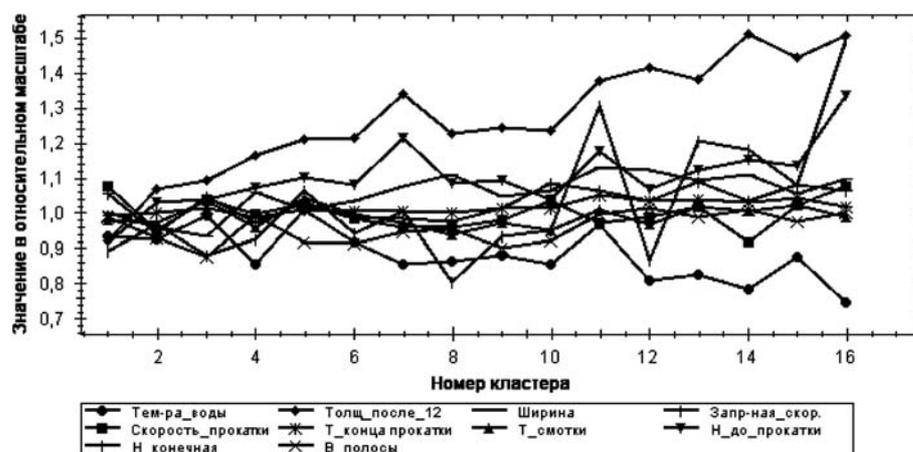


Рис. 2. Иллюстрация поведения значений 10 компонент случайного процесса

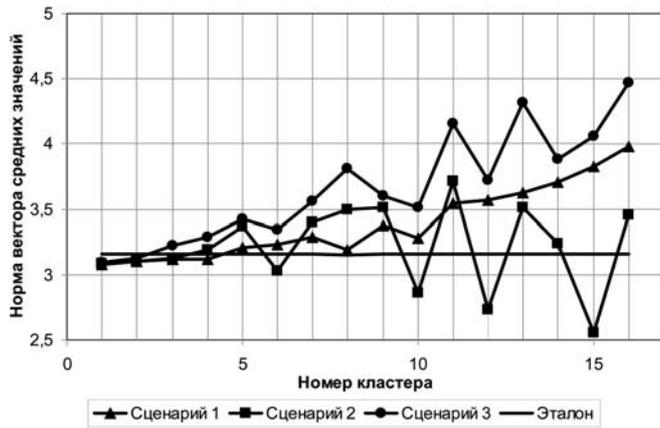


Рис. 3. Изменение характеристики (5) для последовательности кластеров измеренных значений 10-мерного процесса

шей размерности картина усложняется и без свертывания отдельных величин в скалярный показатель, дающий обобщенное представление о контролируемом процессе в целом, трудно оценить состояние и тенденции изменения процесса.

Ситуация принципиально изменяется при использовании евклидовой нормы средних и ковариационной матрицы, обеспечивающих свертку множества реализаций многомерных случайных величин в скалярные статистики. На рис. 3 представлены графики изменения среднего значения 10-мерного векторного процесса, т. е. каждый отдельный график на рис. 3 отражает изменение среднего состояния контролируемого объекта — всю информацию, представленную всеми графиками рис. 2. При этом график "сценарий 1" отражает изменение характеристики (5) при наличии дрейфа среднего состояния объекта. Каждая точка, показанная на нем, получена усреднением по всем 30 измеренным значениям и по всем 10 компонентам случайного вектора соответствующего кластера. График "сценарий 2" на рис. 3 отражает изменение характеристики (5) на выходе системы мониторинга объекта, при нарастании амплитуды отклонений контролируемых значений от средних. График "сценарий 3" отражает изменение среднего показателя объекта, у которого подвержены дрейфу и средние значения величин, и их нестабильность. На графиках рис. 3 достаточно явно выявляется разность в изменении среднего состояния процесса при различных сценариях изменения значений отдельных компонент вектора мониторинга. В частности, график "сценарий 2" на рис. 3, отражающий статистику (5) для объекта, не имеющего линейной составляющей дрейфа (при генерации данных в выражении (11) $\alpha = 0$), показывает отсутствие регулярного изменения евклидовой нормы среднего значения.

На рис. 4 показаны графики изменения энтропии (6), характеризующей доверительную область для контролируемых величин в пространстве мо-

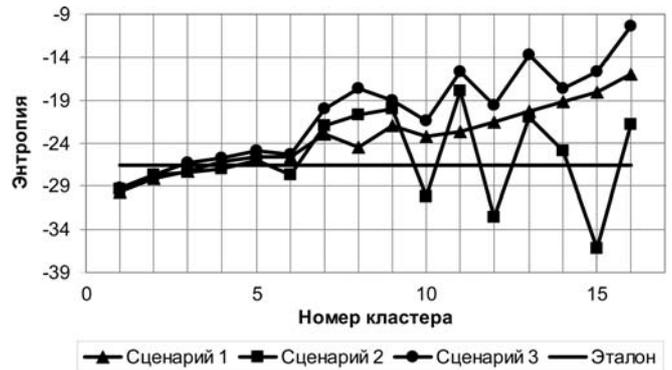


Рис. 4. Изменение характеристики (9) для последовательности кластеров значений 10-мерного процесса

нитинга U . Графики отражают поведение энтропии при рассмотренных вариантах изменения компонент вектора мониторинга. На графике "сценарий 1" энтропия изменяется монотонно вследствие монотонного изменения компонент мониторинга, в то время как график "сценарий 2", отражающий случай нарастания нестабильности (колеблемости) отдельных компонент мониторинга, демонстрирует существенную нестабильность энтропии.

Таким образом, из представленных результатов видно, что характеристики (5) и $R_{ii}(l)$ действительно могут быть использованы в качестве мер оценки среднего состояния объекта и нестабильности его состояния. Они позволяют свернуть информацию, содержащуюся в множествах реализаций векторной случайной величины, в скалярные характеристики среднего состояния и нестабильности объекта.

Заключение

Мониторинг различных объектов и процессов достаточно часто базируется на систематическом измерении множества случайных величин, на основании значений которых необходимо сформировать выводы о состоянии объекта или процесса. В работе предложены статистики, пригодные для оценки состояния объектов по множествам измеренных значений множества случайных величин, контролируемых в процессе мониторинга. Представлена методика их формирования и применения. Приводится иллюстрация оценки среднего значения и нестабильности на примере реального многомерного процесса.

Из содержания разработанных характеристик и иллюстративных примеров видно, что характеристики позволяют свертывать информацию, получаемую при мониторинге многомерных случайных процессов. Важно отметить, что среднее значение и энтропия векторной случайной величины представляют характеристики, усредненные по всем компонентам случайного вектора и по всем его

реализациям, доступным обработке. Содержание технологии мониторинга может быть детализировано введением по аналогии с контрольными картами предупредительных и критических границ, 3σ -х интервалов и т. п. Выход за границы может определяться автоматически соответствующими системами, реализующими предложенные статистики, и иметь соответствующее технологическое развитие.

Разработанные статистические характеристики ориентированы на систематическое применение вычислительных систем и могут быть использованы при синтезе автоматических систем мониторинга.

Список литературы

1. Кузнецов Л. А. Управление качеством в сложных технологических процессах // Проблемы управления. 2007. № 3. С. 47—53.
2. Кузнецов Л. А. Исследование и оценка эволюции технологии // Методы менеджмента качества. 2008. № 3. С. 24—29.
3. Кузнецов Л. А. Некоторые формальные методы выявления технологических причин невыполнения регламента многомерного качества // Приборы и системы: управление, контроль, диагностика. 2010. № 3. С. 1—11.
4. Шеннон К. Математическая теория связи. 1948. — В кн.: К. Шеннон. Работы по теории информации и кибернетике: пер. с англ. под ред. Р. Л. Добрушина и О. Б. Лупанова. М.: ИЛ, 1963.
5. Бард Й. Нелинейное оценивание параметров: пер. с англ. В. С. Дуженко и Е. С. Фоминой // Под ред. и с пред. В. Г. Горского. М.: Статистика, 1979. 349 с.

УДК 004.652:621.396.1

И. А. Иванов, аспирант,
С. У. Увайсов, д-р техн. наук, проф.,
МИЭМ, e-mail: Uvaysov@yandex.ru

Информационная модель процесса проектирования контролепригодных радиоэлектронных средств

Предложена информационная модель процесса обеспечения контролепригодности радиоэлектронных средств на стадии проектирования, представляющая собой комплекс взаимосвязанных модулей для приема, обработки, хранения и передачи данных. В модели заложены все основные свойства и информационные связи объекта при современном автоматизированном проектировании с учетом разнородности протекающих физических процессов.

Ключевые слова: проектирование радиоэлектронных средств; информационная модель; контролепригодность; диагностирование; моделирование электрическое, тепловое, механическое, диагностическое

Проектирование радиоэлектронных средств (РЭС) сопровождается процессом непрерывной передачи, обработки и хранения информации. Для описания и исследования такого рода процессов широко применяются информационные модели.

Информационная модель — модель, которая учитывает существенные свойства объекта и информационные связи между ними и позволяет оценить выходные характеристики процесса проектирования при подаче на модель информации об изменениях входных величин.

Анализ современного состояния процесса автоматизированного проектирования радиоэлектронных средств, а также попыток применения извест-

ных подходов, методов и показателей обеспечения контролепригодности выявил ряд существенных проблем.

Это связано прежде всего с тем, что существующая технология обеспечения контролепригодности ориентирована на оценку достигнутого уровня относительно некоторого аналога, которого, применительно к вновь проектируемым РЭС, может не быть. Кроме того, имеющаяся система противоречивых показателей не позволяет целенаправленно обеспечить контролепригодность электронных средств [2, 3, 6]. Поэтому в работе предложен новый метод автоматизированного проектирования РЭС в аспекте безусловного обеспечения контролепригодности при заданной глубине и требуемой полноте диагностирования технического состояния [7]. В основе метода лежит информационная модель процесса проектирования контролепригодных электронных средств, представленная на рисунке (см. третью сторону обложки).

Разработанная информационная модель может быть описана системой уравнений:

$$\left\{ \begin{array}{l} T_4 = \Phi_1(T_1, T_2) \\ T_5 = \Phi_3(T_1, T_2, T_4) \\ T_6 = \Phi_0(T_1, T_2, T_3) \\ T_7 = \Phi_3(T_2, T_4) \\ T_8 = \Phi_4(T_1, T_2) \\ T_9 = \Phi_5(T_7, T_8) \\ T_{10} = \Phi_6(T_3, T_6, T_9) \\ T_{11} = \Phi_7(T_2, T_3, T_4, T_{12}) \\ T_{12} = \Phi_8(T_2, T_3, T_4, T_{12}) \\ T_{16} = \Phi_{12} \left[\left\{ \begin{array}{l} T_{13} = \Phi_9(T_1, T_4, T_{11}, T_{12}, Y_0) \\ T_{14} = \Phi_{10}(T_1, T_{12}, T_{13}, Y_0) \\ T_{15} = \Phi_{11}(T_1, T_4, T_{12}, T_{14}, Y_0) \end{array} \right\} \right] \\ \left\{ \begin{array}{l} Y_0 = 0, \text{ если } K = 1 \\ Y_0 = \Phi_{13}(T_{16}), \text{ если } K < 1. \end{array} \right. \end{array} \right.$$

Исходной информацией для начала проектирования являются три массива: $T1$, $T2$ и $T3$. Массив данных $T1$ представляет собой техническое задание, содержащее набор требований к проектируемому объекту, а также описание условий его дальнейшей эксплуатации. То есть в данном массиве содержится информация о множестве входных воздействий $X_i(\zeta)$, множестве выходных характеристик $Y_i(\zeta)$, где ζ — независимый аргумент. Массив $T2$ — справочная информация по электрорадиоизделиям (ЭРИ), которая содержит множество внутренних параметров элементов $Q_i(z)$, зависящих от множества внешних воздействий Z [9]. В случае, если проектируемое изделие или его отдельные части имеют аналоги, по которым собрана статистика отказов (рекламаций), то в таком случае формируется массив $T3$, содержащий всю эксплуатационную статистику. $T3$ представляет собой матрицу M_{NV} , где N — число элементов, а V — число параметров применения этих элементов (наработка, интенсивность отказов, причины отказов и т. д.) [8].

Массивы данных $T1$ и $T2$ поступают на вход процедуры формирования математической модели $\Phi1$, в основу которой заложены методы конечных разностей и конечных элементов. Проведение процедуры $\Phi1$ с использованием исходных данных ($T1$, $T2$) позволяет сформировать множество моделей технического процесса ($T4$): $W_i\{X_i(\zeta), Y_i(\zeta), Q_i(z)\}$.

Данное множество представляет собой совокупность моделей различных физических процессов, а именно, электрических, тепловых, механических и др. Например, модель электрических процессов может быть представлена в виде:

$$W_{\Phi}\{[I_{\text{вх}}(\zeta), U_{\text{вх}}(\zeta)], [I_{\text{вых}}(\zeta), U_{\text{вых}}(\zeta)], Q_{\Phi}(z)\} = 0,$$

где $I_{\text{вх}}(\zeta)$, $U_{\text{вх}}(\zeta)$ — множества входных токов и напряжений; $I_{\text{вых}}(\zeta)$, $U_{\text{вых}}(\zeta)$ — множества выходных токов и напряжений; $Q_{\Phi} = Q_{\text{э}} \cup Q_{\text{тф}} \cup Q_{\text{фм}}$ — множество физических параметров, где $Q_{\text{э}}$ — множество внутренних электрических параметров; $Q_{\text{тф}}$ — множество теплофизических параметров; $Q_{\text{фм}}$ — множество физико-механических параметров [5].

С использованием математических моделей физических процессов, протекающих в объекте, проводится электрическое, тепловое и механическое моделирование ($\Phi2$) с применением современных программных средств (MicroCap, OrCad, ТРИАНА, АСОНИКА-ТМ, ANSYS и др.), эффективность которых подтверждается их широким распространением в промышленности.

Результаты моделирования, формирующие массив $T5$, должны удовлетворять всем требованиям технического задания $T1$. $\Phi1$ и $\Phi2$ — это сложные операции, в которых могут протекать многоитерационные процессы по подбору требуемых параметров, элементной базы и элементов конструкций.

$T5$ представляет собой матрицу M_{NP} , где N — число элементов, а P — число расчетных параметров (потенциалы в узлах, токи через элементы, локальные температуры элементов и др.).

Важным звеном в автоматизированном информационном процессе является оператор-эксперт ($\Phi0$). Операции, которые осуществляет эксперт, используя данные $T1$, $T2$, $T3$ и $T4$, позволяют получить массив $T6$, который применяется на дальнейших стадиях контролепригодного проектирования. Обработка входных массивов проводится с помощью имеющегося множества баз знаний E_X [5]:

$$E_X = \{\Pi_1, \Pi_2, \dots, \Pi_i : \Pi_i = \{P_1, P_2, \dots, P_n\}\},$$

где Π_i — i -я база знаний; P_n — множество правил.

Массив $T6$ — мнение эксперта, которое учитывается при формировании набора диагностируемых элементов и представляет собой массивы данных в виде вектор-строки

$$x = \left\{ (x_1, \dots, x_j, \dots, x_n) : \sum_{j=1}^n x_j = 1 \right\},$$

где x_j — весовой коэффициент j -го элемента, по мнению эксперта.

Данная информационная модель описывает в первую очередь, все процессы, связанные с обеспечением контролепригодности. Первым этапом в процессе обеспечения контролепригодности является формирование набора диагностируемых элементов.

Анализируя элементную базу объекта, его схемно-конструкторское решение, технологию проектирования и производства, мы получаем информацию о достаточном наборе элементов ($T10$), обеспечивающем требование технического задания $T1$ по коэффициенту полноты проверки $K_{\text{ПП}}$.

Формирование массива $T10$ осуществляется на основе процедур обработки информации о степени значимости каждого ЭРИ. Для этого операции $\Phi3$ позволяют формировать информацию о чувствительности выходных характеристик к внутренним параметрам элементов ($T7$) в виде матрицы чувствительностей A_{NY} , где N — число параметров элементов; Y — число выходов РЭС. При этом используется множество математических моделей $T4$, описывающих связи выходных характеристик и внутренних параметров $T2$. Используемые методы расчета функции чувствительности достаточно подробно изложены в работе [9].

Параллельно происходит преобразование информационных массивов $T1$, $T2$ и $T3$ в формат входных данных программного комплекса АСОНИКА-К [1] (системы расчета надежности составных частей), где с помощью процедуры $\Phi4$, на основе математических моделей, взятых из справочников по надежности, формируется массив данных $T8$. $T8$ пред-

ставляет собой матрицу, содержащую результаты расчета надежности ЭРЭ.

Процедура (Ф5) обработки массивов $T7$ и $T8$ дает набор весовых коэффициентов, которые приводят множество ЭРИ к ранжированному виду ($T9$) по степеням значимости.

$T9$ — это массив данных в виде упорядоченной вектор-строки:

$$N' = \left\{ (N_i^{r1}, \dots, N_k^{rk}, \dots, N_n^{rn}) : r1 > \dots > rk > \dots > rn, \sum_{k=1}^n rk = 1 \right\},$$

где n — число ЭРИ; rk — значение k -го весового коэффициента, определяющего степень значимости элемента.

Далее проводится выборка (Ф6) из массива $T9$ числа элементов, удовлетворяющего требованиям по полноте проверки. Таким образом формируется массив N'_q . Учет мнения эксперта и имеющихся рекомендаций происходит путем замены последних элементов N'_q на элементы массивов $T3$ и $T6$, в результате чего формируется массив $T10$.

Наборы данных $T1$, $T4$, $T5$ и $T10$ поступают в блок обработки диагностической информации. Для формирования массивов, достаточных для проведения диагностического моделирования, необходимо сформировать список возможных неисправностей и контрольных точек относительно выбранного числа диагностируемых элементов.

В результате процедур обработки входной информации (Ф7 и Ф8) формируются $T11$ и $T12$. $T11$ представляет собой матрицу M_{NQ} , где N — число ЭРИ, а Q — число возможных дефектов. $T12$ представляет собой матрицу M_{RK} , где R — число узлов; K — число контрольных точек. $T12$ соответствует предварительным наборам контрольных точек. В результате диагностического моделирования число контрольных точек и места их расположения могут меняться. Причем, если для электрического моделирования в качестве первоначальных контрольных точек могут быть выбраны выходы схемы, то для теплового моделирования контрольной точкой является корпус каждого ЭРЭ.

Диагностическое моделирование (Ф12) представляет собой многоэтапный информационный процесс, в течение которого осуществляется несколько операций: Ф9, Ф10, Ф11. Для формирования тестовых воздействий ($T13$) обрабатывается информация о возможных дефектах и местах размещения контрольных точек ($T11$, $T12$). Процесс формирования перечня контрольных точек (Ф10)

напрямую связан с набором неисправностей, и с видами и значениями тестовых сигналов. Правильный выбор контрольных точек позволяет повысить эффективность диагностирования.

Основная информационная нагрузка на данном этапе приходится на математическое моделирование физических процессов (Ф11). Процедура Ф11 описывает процесс формирования справочника неисправностей ($T15$), содержащего информацию о видах неисправностей и соответствующих им выходных характеристиках (напряжения в узлах схемы, температура ЭРИ и амплитудно-частотные характеристики в контрольных точках конструкций).

Таким образом, на стадии диагностического моделирования формируется база диагностических данных ($T16$), которая содержит всю информацию о возможных дефектах и методы их идентификации.

Степень однозначности выявления дефектов базой $T16$ определяется процедурой проверки (Ф13) критериальной функции. Если критерий выполняется, то принимается решение о том, что контролепригодность обеспечена. В противном случае, формируется набор рекомендаций в виде управляющего воздействия $U0$. Воздействие позволяет путем целенаправленных изменений параметров процесса диагностического моделирования однозначно обеспечить контролепригодность РЭС.

Таким образом, информационная модель в виде графического и аналитического представлений обеспечивает информационную поддержку процесса автоматизированного контролепригодного проектирования современных радиоэлектронных средств.

Список литературы

1. **Жаднов И. В.** АСОНИКА-К — программный комплекс анализа и обеспечения надежности // Надежность. 2004. № 3(10). С. 5—12.
2. **ГОСТ 23563—79.** Техническая диагностика. Контролепригодность объектов диагностирования. Правила обеспечения. М.: Изд-во стандартов, 1979.
3. **ГОСТ 24029—80.** Техническая диагностика. Категории контролепригодности объектов диагностирования. М.: Изд-во стандартов, 1980.
4. **ГОСТ 26656—85.** Техническая диагностика. Контролепригодность. Общие требования. М.: Изд-во стандартов, 1985.
5. **Гольдин В. В., Журавский В. Г., Сарафанов А. В., Кофанов Ю. Н.** Информационная поддержка жизненного цикла электронных средств. М.: Радио и связь, 2002. 379 с.
6. **Сагунов В. И., Ломакина Л. И.** Контролепригодность структурно связанных систем. М.: Энергоатомиздат, 1990. 111 с.
7. **Увайсов С. У., Увайсов Р. И., Иванов И. А.** Обеспечение контролепригодности радиоэлектронных средств в рамках CALS-технологий // Качество. Инновации. Образование. Вып. 2011. № 68. С. 43—46.
8. **Жаднов В. В., Полесский С. Н., Якубов С. Э.** Оценка качества компонентов компьютерной техники // Надежность. 2008. № 3 (26). С. 26—35.
9. **Кофанов Ю. Н.** Теоретические основы конструирования, технологии и надежности радиоэлектронных средств: учеб. пособие для вузов. М.: Радио и связь, 1991. 360 с.

Д. В. Сперанский, д-р техн. наук, проф.,
МИИТ,
e-mail: speransky.dv@gmail.com

Идентификация состояний одного типа билинейных систем

Объектом исследования являются билинейные системы специального типа. Для систем такого типа получены критерии существования синхронизирующих, установочных и диагностических последовательностей, сформулированные в терминах их характеристических матриц. Предложены и обоснованы аналитические методы поиска перечисленных последовательностей.

Ключевые слова: билинейная система, синхронизирующая последовательность, установочная последовательность, диагностическая последовательность, критерии существования, аналитические методы построения

Введение

Сложность и большая размерность современных цифровых устройств привели к необходимости применения для их проектирования и подготовки тестовых данных в целях контроля и диагностирования специальных программных систем. Эффективная реализация таких систем требует использования передовых информационных технологий. Одна из важнейших функций упомянутых программных систем состоит в моделировании проектируемых и/или реализованных "в железе" цифровых систем.

Известно [1], что отсутствие информации о фазовом состоянии устройства не позволяет предсказывать его реакцию на различные входные воздействия. Очевидно, что в этом случае адекватное моделирование устройства невозможно в принципе. Последний факт приводит к необходимости идентификации (определения) состояния устройства перед началом процесса моделирования его поведения.

Существующие методы идентификации состояния устройства с памятью можно разделить на аппаратные и программные. Аппаратные методы применимы только для реализованных "в железе" устройств. Их можно считать эквивалентными наличию в устройстве кнопки сброса, "обнуляющей" все регистры и другие элементы памяти. К сожалению, этот метод не гарантирует установки устройства в требуемое начальное состояние, если само устройство сброса оказывается неисправным. Альтернативный, но являющийся по существу универсальным, метод установки устройства — программный, предполагает подачу специальных входных

последовательностей. Понятно, что таким методом можно моделировать и аппаратную установку.

В предлагаемой статье рассматриваются проблемы существования и разработки методов поиска различных типов, идентифицирующих состояние входных последовательностей, ориентированных на устройства, математическими моделями которых являются билинейные автоматы специального типа. Заметим, что решение названных проблем является необходимым условием возможности применения программных систем проектирования и/или синтеза тестов для любых типов цифровых устройств с памятью.

Основные определения и постановка задач

Начнем с краткого описания модели билинейного автомата (БА), которая используется в статье, а более детальное описание ее приведено в работе [2]. БА является системой с конечным числом входных полюсов и конечным числом выходных полюсов, на которых наблюдаются реакции системы. Воздействия, поступающие на БА, прикладываются одновременно ко всем входам в дискретные моменты времени, которые для удобства представляются целыми числами.

Условимся, что число входных полюсов равно l , а число выходных полюсов равно n . Предполагается, что входные сигналы принимают значения из поля $GF(p) = \{0, 1, \dots, p-1\}$, где p — простое число. Под состоянием БА понимается упорядоченная совокупность состояний элементов задержек, входящих в структуру БА. Множество состояний БА обозначим через S_n .

Введем следующие обозначения:

$$\mathbf{u}(t) = [u_1(t), \dots, u_l(t)]^T, \mathbf{s}(t) = [s_1(t), \dots, s_n(t)]^T, \\ \mathbf{y}(t) = [y_1(t), \dots, y_n(t)]^T.$$

Здесь $\mathbf{u}(t)$, $\mathbf{s}(t)$, $\mathbf{y}(t)$ — входной вектор, вектор состояния и выходной вектор соответственно.

Функционирование БА задается следующими системами уравнений переходов и выходов:

$$\mathbf{s}(t+1) = \mathbf{A}\mathbf{s}(t) + \left(\sum_{i=1}^l \mathbf{F}_i u_i(t) \right) + \mathbf{B}\mathbf{u}(t), \quad (1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t) + \mathbf{D}\mathbf{u}(t), \quad (2)$$

где \mathbf{A} , \mathbf{F}_i — матрицы размерности $n \times n$; \mathbf{B} — матрица размерности $n \times l$; \mathbf{C} — матрица размерности $m \times n$, \mathbf{D} — матрица размерности $m \times l$.

Упомянутые матрицы называются характеристическими матрицами БА. Элементы этих матриц являются элементами поля $GF(p)$.

Введем следующее обозначение:

$$I(\mathbf{u}(t)) = \sum_{i=1}^l \mathbf{F}_i u_i(t).$$

Пусть задана входная последовательность $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(t)$ и пусть $\mathbf{s}(0)$ есть начальное состояние БА. Методом математической индукции можно доказать, что конечное состояние БА после подачи упомянутой последовательности вычисляется по формуле

$$\begin{aligned} \mathbf{s}(t+1) = & \left(\prod_{i=0}^t (\mathbf{A} + I(\mathbf{u}(t-i))) \right) \mathbf{s}(0) + \\ & + \left(\prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(t-i))) \right) \mathbf{B}\mathbf{u}(0) + \dots + \\ & + (\mathbf{A} + I(\mathbf{u}(t)))\mathbf{B}\mathbf{u}(0) + \mathbf{B}\mathbf{u}(t). \end{aligned} \quad (3)$$

Аналогично доказывается формула для вычисления реакции БА на ту же последовательность:

$$\begin{aligned} \mathbf{y}(t) = & \mathbf{C} \left[\prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \mathbf{s}(0) + \right. \\ & + \prod_{i=0}^{t-2} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \mathbf{B}\mathbf{u}(0) + \\ & + \prod_{i=0}^{t-3} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \mathbf{B}\mathbf{u}(1) + \dots + \\ & \left. + \mathbf{B}\mathbf{u}(t-1) \right] + \mathbf{D}\mathbf{u}(t). \end{aligned} \quad (4)$$

Далее условимся в следующем:

а) если в формулах (3) и (4) значение $t < 0$, то полагаем $I(t) = [0]$, где $[0]$ — нулевая матрица соответствующей размерности;

б) если в произведении $\prod_{i=0}^k f(i)$ значение $k < 0$,

то все это произведение полагаем равным \mathbf{E} — единичной матрице соответствующей размерности.

Рассматриваемые ниже задачи идентификации БА будем решать путем подачи на его вход подходящей последовательности. Далее экспериментом с БА будем называть процесс подачи на него последовательности входных сигналов, наблюдения реакции на эту последовательность и вывода заключений о его фазовом состоянии. Определим теперь различные типы экспериментов, которые мы будем исследовать.

Синхронизирующий эксперимент требует наличия синхронизирующей последовательности (СП), которая переводит БА в одно и то же состояние независимо от того, из какого состояния он стартовал.

Установочный эксперимент проводится с использованием установочной последовательности (УП), которая по наблюдаемой реакции на нее однозначно идентифицирует конечное состояние БА.

Диагностический эксперимент проводится с использованием диагностической последовательности (ДП), которая по наблюдаемой реакции на нее однозначно идентифицирует состояние БА, в котором он находился перед подачей ДП.

Очевидно СП можно рассматривать как вырожденный случай УП, поскольку подача СП вызывает перевод БА в известное конечное состояние, хотя и не требует при этом наблюдения его реакции.

Отметим также, что каждая ДП одновременно является и УП, поскольку, определив по реакции БА на ДП его начальное состояние, легко определить по формуле (3) состояние, в котором он окажется после подачи ДП. Обратное утверждение, однако, неверно, т. е. не каждая УП является одновременно и ДП.

Сформулируем теперь два типа задач, исследуемых в статье.

1. Для БА, заданных уравнениями (3) и (4), найти условия существования СП, УП и ДП.

2. Разработать аналитические методы поиска СП, УП и ДП, используемых для идентификации состояний заданных БА.

Синхронизирующие последовательности

В работе [2] был рассмотрен БА, отличающийся от рассматриваемого в нашей статье иного вида функцией выходов, но с той же функцией переходов. Поскольку идентификация состояния с использованием СП не предполагает наблюдения реакций БА, то полученные в работе [2] результаты оказываются справедливыми и для нашего случая. Так, имеет место следующее утверждение.

Утверждение 1. Для того чтобы входная последовательность $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(t)$ была СП для БА, необходимо и достаточно, чтобы выполнялось равенство

$$\prod_{i=0}^t (\mathbf{A} + I(\mathbf{u}(t-i))) = [0]. \quad (5)$$

Следствие. Для того чтобы входная последовательность $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(t)$ была синхронизирующей последовательностью для БА, достаточно, чтобы по крайней мере для одного из значений $i = 0, 1, \dots, t$ выполнялось равенство

$$(\mathbf{A} + I(\mathbf{u}(t-i))) = [0]. \quad (6)$$

Приведем еще одно достаточное условие существования СП, предварительно дав следующее определение. Назовем квадратную матрицу верхней (нижней) треугольной, если все ее элементы, лежащие на главной диагонали и ниже (выше) ее, нулевые.

УП совпадают, то конечные состояния этого БА также совпадают.

Для БА типа, исследуемого в статье, с учетом формул (3) и (4), приведенное высказывание в формализованном виде принимает следующий вид:

$$\begin{aligned} \forall \mathbf{s}_1, \mathbf{s}_2 \in S_n \bigwedge_{d=0}^t \left\{ \mathbf{C} \left[\prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \mathbf{s}_1 + \right. \right. \\ \left. \left. + \prod_{i=0}^{t-2} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \mathbf{B}\mathbf{u}(0) + \dots + \mathbf{B}\mathbf{u}(t-1) \right] + \right. \\ \left. + \mathbf{D}\mathbf{u}(t) = \mathbf{C} \left[\prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \mathbf{s}_2 + \prod_{i=0}^{t-2} (\mathbf{A} + \right. \right. \\ \left. \left. + I(\mathbf{u}(t-1-i))) \mathbf{B}\mathbf{u}(0) + \dots + \mathbf{B}\mathbf{u}(t-1) \right] + \mathbf{D}\mathbf{u}(t) \right\} \rightarrow \\ \rightarrow = \left[\prod_{i=0}^t (\mathbf{A} + I(\mathbf{u}(t-i))) \mathbf{s}(0) + \left(\prod_{i=0}^{t-1} (\mathbf{A} + \right. \right. \\ \left. \left. + I(\mathbf{u}(t-i))) \right) \mathbf{B}\mathbf{u}(0) + \dots + (\mathbf{A} + I(\mathbf{u}(t))) \mathbf{B}\mathbf{u}(0) + \right. \\ \left. + \mathbf{B}\mathbf{u}(t) \right] = \left[\prod_{i=0}^t (\mathbf{A} + I(\mathbf{u}(t-i))) \mathbf{s}_1 + \left(\prod_{i=0}^{t-1} (\mathbf{A} + \right. \right. \\ \left. \left. + I(\mathbf{u}(t-i))) \right) \mathbf{B}\mathbf{u}(0) + \dots + (\mathbf{A} + I(\mathbf{u}(t))) \mathbf{B}\mathbf{u}(0) + \right. \\ \left. + \mathbf{B}\mathbf{u}(t) \right] = \left[\prod_{i=0}^t (\mathbf{A} + I(\mathbf{u}(t-i))) \mathbf{s}_2 + \left(\prod_{i=0}^{t-1} (\mathbf{A} + \right. \right. \\ \left. \left. + I(\mathbf{u}(t-i))) \right) \mathbf{B}\mathbf{u}(0) + \dots + (\mathbf{A} + I(\mathbf{u}(t))) \mathbf{B}\mathbf{u}(0) + \mathbf{B}\mathbf{u}(t) \right]. \end{aligned}$$

Знаки $\bigwedge_{d=0}^t$ и $\bigvee_{d=0}^t$ здесь и далее означают дизъюнкцию и конъюнкцию соответственно выражений, стоящих за этими знаками.

Преобразуя каждое в отдельности равенство в приведенном высказывании путем переноса правых их частей в левые и осуществив соответствующие сокращения, получим

$$\begin{aligned} \forall \mathbf{s}_1, \mathbf{s}_2 \in S_n \left[\bigwedge_{d=0}^t \left\{ \mathbf{C} \left[\prod_{i=0}^{d-1} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \right] \times \right. \right. \\ \left. \left. \times (\mathbf{s}_1 - \mathbf{s}_2) \right\} \rightarrow \left[\prod_{i=0}^t (\mathbf{A} + I(\mathbf{u}(t-i))) \right] (\mathbf{s}_1 - \mathbf{s}_2) = [0]. \right. \end{aligned}$$

Поскольку \mathbf{s}_1 и \mathbf{s}_2 произвольные не совпадающие между собой состояния, то и состояние $(\mathbf{s}_1 - \mathbf{s}_2)$ также может быть любым состоянием из S_n . Таким образом, в последнем выражении разность $(\mathbf{s}_1 - \mathbf{s}_2)$ можно заменить на произвольное состояние \mathbf{s} , отличное от нулевого. Из проделанных выкладок с учетом эквивалентности высказывания $(a \rightarrow b) \equiv \equiv (\bar{a} \vee b)$ следует справедливость следующего утверждения.

Утверждение 3. Для того чтобы последовательность $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(t)$ была УП для БА, необходимо и достаточно, чтобы для каждого ненулевого состояния из S_n выполнялось по крайней мере одно из двух следующих условий:

- 1) $\bigvee_{d=0}^t \left[\mathbf{C} \left[\prod_{i=0}^{d-1} (\mathbf{A} + I(\mathbf{u}(t-1-i))) \right] \mathbf{s} \neq [0]; \right.$
- 2) $\prod_{i=0}^{d-1} (\mathbf{A} + I(\mathbf{u}(t-i))) \mathbf{s} = [0].$

Заметим, что второе условие есть условие существования СП (см. *утверждение 1*). В связи с этим напомним, что СП есть вырожденный случай УП.

Для поиска УП воспользуемся идеей построения установочного дерева, детально описанного в работе [1]. Напомним, что при построении такого дерева проводится перебор различных входных последовательностей в целях определения среди них УП. По существу такой поиск сводится к проверке, может ли анализируемая конкретная входная последовательность быть УП. В работе [1] установочное дерево представляется в виде графа. Ниже будет показано на конкретном примере, что всю необходимую информацию для анализа любой последовательности, извлекаемую из графа, всегда можно получить предложенным нами аналитическим методом.

Пусть задан БА над полем $GF(2)$ следующими характеристическими матрицами:

$$\begin{aligned} \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 10 \\ 01 \\ 10 \end{bmatrix}, \mathbf{F}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{F}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \\ \mathbf{C} = \begin{bmatrix} 110 \\ 001 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 10 \\ 01 \end{bmatrix}. \end{aligned}$$

Предполагается, что множество допустимых начальных состояний БА совпадает со множеством всех его состояний. Исследуем, например, входную последовательность $\mathbf{u}(0) = [11]^T$, $\mathbf{u}(1) = [1 \ 1]^T$ и проверим, является ли она УП для рассматриваемого БА.

Легко убедиться, что функции переходов и выходов рассматриваемого БА имеют следующий вид:

$$\mathbf{s}(t+1) = \mathbf{A}\mathbf{s}(t) + \left(\sum_{i=1}^l \mathbf{F}_i u_i(t) \right) + \mathbf{B}\mathbf{u}(t) =$$

$$= \begin{bmatrix} s_1(t) + s_3(t)(u_1(t) + u_2(t)) + u_1(t) \\ s_2(t)(u_1(t) + 1) + u_2(t) \\ s_1(t)u_2(t) + s_3(t)(u_2(t) + 1) + u_1(t) \end{bmatrix}; \quad (8)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{s}(t) + \mathbf{D}\mathbf{u}(t) = \begin{bmatrix} s_1(t) + s_2(t) + u_1(t) \\ s_3(t) + u_2(t) \end{bmatrix}. \quad (9)$$

Рассмотрим все четыре гипотетически возможные реакции $\mathbf{y}(0)$, вход $\mathbf{u}(0) = [1 \ 1]^T$ и определим те состояния БА, из которых они могут последовать. Очевидно, что эти состояния можно найти как решения следующих систем уравнений:

$$1) \mathbf{y}(0) = \begin{bmatrix} s_1(0) + s_2(0) + 1 \\ s_3(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$2) \mathbf{y}(0) = \begin{bmatrix} s_1(0) + s_2(0) + 1 \\ s_3(0) + 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$3) \mathbf{y}(0) = \begin{bmatrix} s_1(0) + s_2(0) + 1 \\ s_3(0) + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

$$4) \mathbf{y}(0) = \begin{bmatrix} s_1(0) + s_2(0) + 1 \\ s_3(0) + 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Легко проверить, что каждая из этих систем имеет по два решения:

$$1) \mathbf{s}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}; \quad 2) \mathbf{s}(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix};$$

$$3) \mathbf{s}(0) = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}; \quad 4) \mathbf{s}(0) = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}.$$

Используя формулы (8) и (9), легко убедиться в следующем:

а) из состояний $[000]^T$, $[001]^T$, $[010]^T$, $[011]^T$ БА переходит в одно и то же состояние $[111]^T$ и выдает соответственно реакции $[11]^T$, $[10]^T$, $[01]^T$, $[00]^T$;

б) из состояний $[100]^T$, $[101]^T$, $[110]^T$, $[111]^T$ БА также переходит в одно и то же состояние $[010]^T$ и выдает соответственно реакции $[01]^T$, $[00]^T$, $[11]^T$, $[10]^T$.

Приведенные данные позволяют сделать вывод, что входной сигнал $[11]^T$ не является УП. Действительно, например, наблюдаемая реакция $[10]^T$ может

соответствовать двум разным конечным состояниям: $[111]^T$ и $[010]^T$. Заметим, что тот же самый вывод может быть получен и из проверки условий *утверждения* 3. Легко убедиться, что они оба не выполняются.

Перейдем теперь к рассмотрению входной последовательности длины 2 $\mathbf{u}(0) = [11]^T$, $\mathbf{u}(1) = [11]^T$. Поскольку после подачи сигнала $\mathbf{u}(0)$ БА может оказаться только в двух состояниях — $[010]^T$ или $[111]^T$, то они будут являться стартовыми при подаче очередного сигнала $\mathbf{u}(1) = [11]^T$. Используя вновь формулы (8) и (9), вычислим конечные состояния БА и его реакции. Полученные данные о поведении БА при подаче на его вход $\mathbf{u}(0) = [11]^T$, $\mathbf{u}(1) = [11]^T$ сведем в таблицу.

Приведенные в таблице данные свидетельствуют о том, что входная последовательность $\mathbf{u}(0) = [11]^T$, $\mathbf{u}(1) = [11]^T$ является УП для заданного БА. Действительно, любая реакция, из приведенных в таблице, конечное состояние БА определяет однозначно.

Начальное состояние	Выходная реакция	Конечное состояние	Начальное состояние	Выходная реакция	Конечное состояние
$[000]^T$	$[11], [10]$	$[010]^T$	$[100]^T$	$[01], [01]$	$[111]^T$
$[001]^T$	$[10], [10]$	$[010]^T$	$[101]^T$	$[00], [01]$	$[111]^T$
$[010]^T$	$[01], [10]$	$[010]^T$	$[110]^T$	$[11], [01]$	$[111]^T$
$[011]^T$	$[00], [10]$	$[010]^T$	$[111]^T$	$[10], [01]$	$[111]^T$

Диагностические последовательности

Приведенное нами ранее определение того, что последовательность $\mathbf{u}(0)$, $\mathbf{u}(1)$, ..., $\mathbf{u}(t)$ является диагностической последовательностью эквивалентно следующему высказыванию: из равенства двух наблюдаемых на ДП реакций БА следует, что перед подачей ДП этот автомат находился в одном и том же начальном состоянии. В формализованном виде с учетом формул (3) и (4) это высказывание принимает следующий вид:

$$\forall \mathbf{s}_1, \mathbf{s}_2 \in S_n \bigwedge_{d=0}^t \left\{ \mathbf{C} \left[\prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(d-1-i))) \mathbf{s}_1 + \dots + \mathbf{C}\mathbf{B}\mathbf{u}(d-1) \right] + \mathbf{D}\mathbf{u}(d) = \right.$$

$$= \left. \mathbf{C} \left[\prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(d-1-i))) \mathbf{s}_2 + \dots + \mathbf{C}\mathbf{B}\mathbf{u}(d-1) \right] + \mathbf{D}\mathbf{u}(d) \right\} \rightarrow \mathbf{s}_1 \equiv \mathbf{s}_2.$$

Выполнив преобразования, аналогичные тем, что были осуществлены при доказательстве *утверждения 3*, получим высказывание

$$\forall \mathbf{s}_1, \mathbf{s}_2 \in S_n \bigwedge_{d=0}^t \left\{ \mathbf{C} \prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(d-1-i))) \times \right. \\ \left. \times (\mathbf{s}_2 - \mathbf{s}_1) = [0] \right\} \rightarrow \mathbf{s}_1 \equiv \mathbf{s}_2.$$

Если последовательность $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(t)$ для БА является диагностической, то знание реакции на нее позволяет однозначно идентифицировать начальное состояние БА. С учетом формулы (4) реакции БА можно представить как функции от его начального состояния $\mathbf{s}(0)$:

$$\begin{aligned} \mathbf{y}(0) &= \mathbf{C}\mathbf{s}(0) + \mathbf{D}\mathbf{u}(0), \\ \mathbf{y}(1) &= \mathbf{C}(\mathbf{A} + I(\mathbf{u}(0))\mathbf{s}(0) + \mathbf{C}\mathbf{B}\mathbf{u}(0) + \mathbf{D}\mathbf{u}(1), \\ &\dots \\ \mathbf{y}(t) &= \mathbf{C} \prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(t-1-i)))\mathbf{s}(0) + \dots + \mathbf{D}\mathbf{u}(t). \end{aligned}$$

Поскольку ДП и реакции на нее известны, выписанные равенства можно представить в виде

$$\begin{aligned} \mathbf{C}\mathbf{s}(0) &= \Phi_0(\mathbf{u}(0)), \\ \mathbf{C}(\mathbf{A} + I(\mathbf{u}(0))\mathbf{s}(0) &= \Phi_1(\mathbf{u}(0), \mathbf{u}(1)), \\ &\dots \\ \mathbf{C} \prod_{i=0}^{t-1} (\mathbf{A} + I(\mathbf{u}(t-1-i)))\mathbf{s}(0) &= \Phi_t(\mathbf{u}(0), \dots, \mathbf{u}(t)), \end{aligned}$$

где $\Phi_i(\mathbf{u}(0), \dots, \mathbf{u}(i)), i = 0, 1, \dots, t$, — некоторые значения из поля $GF(p)$.

Эти соотношения можно интерпретировать как систему линейных уравнений относительно координат вектора $\mathbf{s}(0)$. Однозначность идентификации состояния $\mathbf{s}(0)$ при диагностическом эксперименте означает, что выписанная система должна иметь единственное решение. Для этого необходимо и достаточно, чтобы ранг этой системы был равен числу ее неизвестных [3]. Отсюда вытекает справедливость следующего утверждения.

Утверждение 4. Для того чтобы последовательность $\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(t)$ была ДП для БА размерности n , необходимо и достаточно, чтобы ранг следующей матрицы был равен n :

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{C}[\mathbf{A} + I(\mathbf{u}(0))] \\ \dots \\ \mathbf{C} \prod_{i=0}^{t-1} [\mathbf{A} + I(\mathbf{u}(t-1-i))] \end{bmatrix}.$$

Приведенная матрица носит название диагностической.

Заметим, что для поиска ДП можно воспользоваться диагностическим деревом, описанным в работе [1]. Как и в случае УП, поиск по этому дереву эквивалентен полному перебору различных входных последовательностей. Для каждой конкретной входной последовательности осуществляется анализ того, является ли она диагностической. Если в работе [1] этот анализ проводится на графовой модели, то *утверждение 4* позволяет выполнить его аналитически. Проиллюстрируем сказанное на примере БА, описанном в предыдущем разделе.

Вновь рассмотрим последовательность $\mathbf{u}(0) = [11]^T, \mathbf{u}(1) = [11]^T$. Для последовательности $\mathbf{u}(0)$ длины 1 диагностическая матрица совпадает с матрицей \mathbf{C} . Из работы [3] известно, что в нашем примере ранг матрицы не может превышать 2, т. е. $\mathbf{u}(0)$ не может быть ДП. Действительно, как следует из приведенных в предыдущем разделе данных, например, наблюдаемая реакция $[10]^T$ может последовать из двух разных начальных состояний ($[000]^T$ или $[011]^T$).

Для входной последовательности $\mathbf{u}(0), \mathbf{u}(1)$ длины 2 диагностическая матрица рассматриваемого БА такова:

$$\begin{bmatrix} \mathbf{C} \\ \mathbf{C}(\mathbf{A} + I(\mathbf{u}(0))) \end{bmatrix} = \begin{bmatrix} 110 \\ 001 \\ 100 \\ 100 \end{bmatrix}.$$

Поскольку ранг последней матрицы равен 3, то в соответствии с *утверждением 4* анализируемая последовательность является ДП. В этом легко убедиться по данным в представленной выше таблице. Таким образом, анализируемая нами входная последовательность для рассматриваемого БА одновременно является и УП, и ДП.

Заключение

Отметим, что методы проверки условий существования УП и ДП, приведенные в работе [1], требуют построения очень громоздких конструкций — установочных и диагностических деревьев. Полученные в статье результаты показывают, что критерии существования СП, УП и ДП для БА удается сформулировать в терминах характеристических матриц БА. Эти аналитические критерии конструктивно проверяемы и их проверка не представляет особой сложности.

Список литературы

1. Гилл А. Введение в теорию конечных автоматов. — М.: Наука, 1966. 272 с.
2. Сперанский Д. В. Лекции по теории экспериментов с конечными автоматами. — М.: Интернет-Университет информационных технологий: БИНОМ. Лаборатория знаний, 2010. 287 с.
3. Курош А. Г. Курс высшей алгебры. — М.: Наука, Главная редакция физ.-мат. лит., 1976. 279 с.

УДК 004.82

С. В. Фролов, д-р техн. наук, проф., зав. каф.,
А. В. Горбунов, д-р мед. наук, проф.,
М. А. Лядов, магистрант,
С. Г. Фареев, аспирант,
 Тамбовский государственный
 технический университет,
 e-mail: Sergej.Frolov@gmail.com

Блочно-иерархический подход для построения базы знаний информационной системы мониторинга состояния головного мозга человека

Для прогнозирования критических состояний сложных объектов в информационной системе мониторинга предложено использовать базы знаний, построенные на основе аппарата булевой алгебры. Проблема большой размерности таблиц истинности, на основе которых организуются знания, решается с применением блочно-иерархического подхода. Построена база знаний информационной системы мониторинга состояния головного мозга человека для прогнозирования угрозы возникновения мозгового инсульта на стадии, когда возможно эффективное использование профилактических мероприятий.

Ключевые слова: булева алгебра, блочно-иерархический подход, база знаний, информационная система мониторинга, сложная система, критическое состояние объекта, состояние головного мозга человека, мозговой инсульт

Введение

На практике часто возникают проблемы при работе с объектами, которые представляют собой сложные системы [1]. Примерами сложных объектов могут быть административно-территориальные образования, организации, транспортные системы, технологическое оборудование, технологические процессы, экологические, биологические объекты и др. Особую значимость представляет прогнозирование состояний, при которых дальнейшее нормальное функционирование объекта невозможно. Состояния, возникающие при нарушении работы одной из подсистем и делающие невозможным функционирование сложной системы в целом, яв-

ляются критическими. Актуальность приобретает одновременная работа с множеством однотипных сложных объектов. Своевременное прогнозирование критических состояний для каждого из множества однотипных объектов можно проводить с использованием информационной системы мониторинга (ИСМ), которая обеспечивает систематическое наблюдение за состоянием подсистем каждого объекта в целях их оценки, контроля или прогноза. По входным параметрам на основе базы знаний (БЗ) ИСМ можно прогнозировать критические состояния объекта мониторинга. Знания могут быть организованы в виде таксономий, онтологий, семантических сетей, продукций, хромосом, фреймов, паттернов [2]. Во многих случаях входные параметры и выходные мониторинговые показатели ИСМ представляются в виде булевых переменных. В этом случае знания могут быть организованы в виде таблиц истинности. Как правило, при мониторинге сложных объектов имеется большое число входных параметров. Например, таблица истинности для 32 входных параметров будет иметь более 10^9 строк, заполнить которую экспертом неосуществимо. Преодоление трудностей при построении БЗ с использованием таблиц истинности возможно при применении положений блочно-иерархического подхода (БИП), позволяющего успешно справиться с проблемами сложности при проектировании систем [3, 4].

Информационная и процедурная модель для системы мониторинга оценки критического состояния объекта

На рис. 1 показана сложная система S как объект, за состоянием которого ведется мониторинг. Сис-

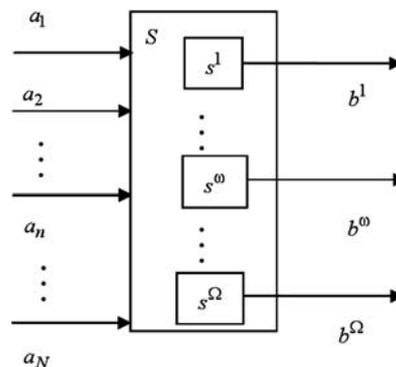


Рис. 1. Система S как объект мониторинга

тема S состоит из подсистем: $S = \{s^1, s^2, \dots, s^\omega, \dots, s^\Omega\}$. Множество $A = \{a_1, a_2, \dots, a_n, \dots, a_N\}$ соответствует множеству входных параметров, которые можно измерять на объекте. Множество значений параметра a_n , $n = \overline{1, N}$, есть $\{0, 1, -\}$: $a_n = 0$ — норма n -го параметра; $a_n = 1$ — параметр n выходит за границу нормы. Если параметр n является неопределенным, то $a_n = -$ и параметр n относится к множеству N входных неопределенных параметров. Объект S определяется состоянием его подсистем s^ω . Здесь $\omega = \overline{1, \Omega}$, где ω — номер подсистемы; Ω — число подсистем в системе S .

Состояние подсистемы s^ω характеризуется выходным мониторинговым показателем b^ω . Здесь $b^\omega = 0$ означает, что состояние подсистемы s^ω не критическое, $b^\omega = 1$ — состояние подсистемы s^ω критическое. Функционирование системы S невозможно, если хотя бы одна из подсистем s^ω находится в критическом состоянии, т. е. выполняется условие

$$\bigvee_{\omega=0}^{\Omega} b^\omega = 1. \text{ Актуальной задачей является создание}$$

системы мониторинга оценки возможности возникновения опасных для жизни заболеваний. Тогда система S представляет собой конкретного человека, а подсистема s^ω характеризует возможность возникновения опасного для жизни заболевания ω .

Состояние b^ω определяется оператором в зависимости от значений множества A :

$$b^\omega = O^\omega(A). \quad (1)$$

Здесь O^ω — оператор, представляющий процедурную модель определения критического состояния подсистемы s^ω , он реализуется ИСМ.

Основой ИСМ является реализация оператора (1), правила работы с которым одинаковы для всех подсистем ω , ($\omega = \overline{1, \Omega}$). Оператору (1) соответствует булева функция

$$b = F(a_1, a_2, \dots, a_p, \dots, a_N), \quad (2)$$

которая определяется таблицей истинности с числом строк равным 2^n . Булевы переменные a_n ($n = \overline{1, N}$) образуют множество A , $a_n \in A$, и характеризуют "норму" входных параметров. Таблица истинности (2) строится с учетом того, что все переменные a_n определены, т. е. a_n принимает только значения из множества $\{0, 1\}$. Неопределенность параметра $a_n = -$ в БЗ не рассматривается.

Для решения проблемы заполнения экспертом таблицы истинности с большим числом строк используются положения БИП [3, 4]. Согласно методу БИП создается иерархическая система блоков показателей, которую удобно представить в виде 4-уров-

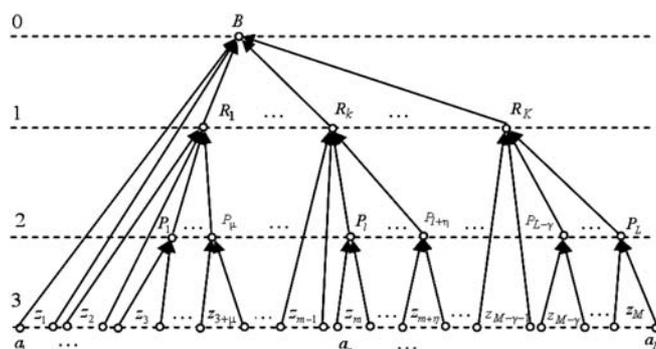


Рис. 2. Иерархическая система блоков показателей БЗ ИСМ

невой структуры (рис. 2). Вводится понятие: блок показателей, представляющий в общем случае булеву функцию, которая определяется таблицей истинности. Блоки показателей располагаются на уровнях 2, 1, 0 иерархии. Набор входных параметров блока показателей формируется из условий конкретной задачи. Выходной показатель блока может быть входным параметром для другого блока показателей.

Верхний уровень (уровень 0) всегда представляется единственным выходным мониторинговым показателем b , нижний уровень (уровень 3) — множеством $A = \{a_1, a_2, \dots, a_n, \dots, a_N\}$, характеризующем входные параметры. Задается разбиение множества A аргументов исходной булевой функции $F(A)$ на подмножества Z_m ($m = \overline{1, M}$):

$$A = Z_1 \cup Z_2 \dots \cup Z_m \cup \dots \cup Z_M, Z_i \cap Z_j = \emptyset, i \neq j. \quad (3)$$

На уровне 2 формируются показатели $p_l = P_l(Z_m)$, где P_l ($l = \overline{1, L}$) — булева функция, которая называется блоком l показателей уровня 2. На уровне 1 формируются показатели $r_k = R_k(P_l(Z_m), \dots, P_{l+\eta}(Z_{m+\eta}), Z_{m-1})$, где R_k ($k = \overline{1, K}$) — булева функция, являющаяся k -м блоком показателей уровня 1. Таким образом, булева функция (2) представляется в виде суперпозиции булевых функций:

$$b = B(R_1(P_1(Z_3), \dots, P_\mu(Z_{3+\mu})Z_2), \dots, R_k(P_l(Z_m), \dots, P_{l+\eta}(Z_{m+\eta}), Z_{m-1}), \dots, R_L(P_{L-\gamma}(Z_{M-\gamma}), \dots, P_L(Z_M), Z_{M-\gamma-1}), Z_1). \quad (4)$$

Декомпозиция булевой функции (2) проведена по методу многоблочной разделительной декомпозиции [5].

Булевой функции (4) соответствует граф в виде дерева (см. рис. 2), которое представляется множеством:

$$\Gamma = \{C, T, D, V\}. \quad (5)$$

Элементами множества Γ являются матрицы, определяющие структуру декомпозиции булевой функции F . Множество Γ будем называть шаблоном иерархии блоков показателей. Здесь $C = \|c_n\|$,

$n = \overline{1, N}$; $c_n = 1$, если a_n — смежная с вершиной B , $c_n = 0$ в противном случае; $\mathbf{T} = \|t_{k,l}\|$, $k = \overline{1, K}$; $l = \overline{1, L}$; $t_{k,l} = 1$, если вершина R_k смежная с P_l , $t_{k,l} = 0$ в противном случае. Для $\mathbf{D} = \|d_{k,n}\|$, $\mathbf{V} = \|v_{l,n}\|$, $d_{k,n} = v_{l,n} = 1$, если вершины R_k, P_l смежные с a_n , $d_{k,n} = v_{l,n} = 0$ в противном случае.

Разбиение (3) множества A на непересекающиеся подмножества Z_m , ($m = \overline{1, M}$) задается условием:

$$c_n + d_{k,i} + d_{u,j} + v_{l,\gamma} + v_{\lambda,\eta} = 1, \quad (6)$$

$$n, i, j, \gamma, \eta = \overline{1, N}; k, u = \overline{1, K}; l, \lambda = \overline{1, L}; \\ i \neq j, \gamma \neq \eta, u \neq k, l \neq \lambda.$$

Упорядоченное множество Π определяет нормальные значения входных параметров и представляется последовательностью N кортежей:

$$\Pi = \{ \langle \alpha_1, \beta_1, \gamma_1, \delta_1, [\underline{\varepsilon}_1, \overline{\varepsilon}_1] \rangle, \dots, \\ \dots, \langle \alpha_n, \beta_n, \gamma_n, \delta_n, [\underline{\varepsilon}_n, \overline{\varepsilon}_n] \rangle, \dots \}, \quad (7)$$

где α_n — целое число, соответствующее названию n -го параметра; β_n — тип значения n -го параметра; $\beta_n = 1$ — логический тип; $\beta_n = 2$ — целый (количественный) тип; $\beta_n = 3$ — интервальный тип. Множество значений элемента γ_n есть $\{0, 1\}$. Здесь γ_n характеризует норму n -го параметра, когда $\beta_n = 1$. Значение $\gamma_n = 0$ соответствует понятию "отсутствует", $\gamma_n = 1$ — понятию "имеется". Элемент δ_n принимает целые числа и соответствует норме n -го параметра

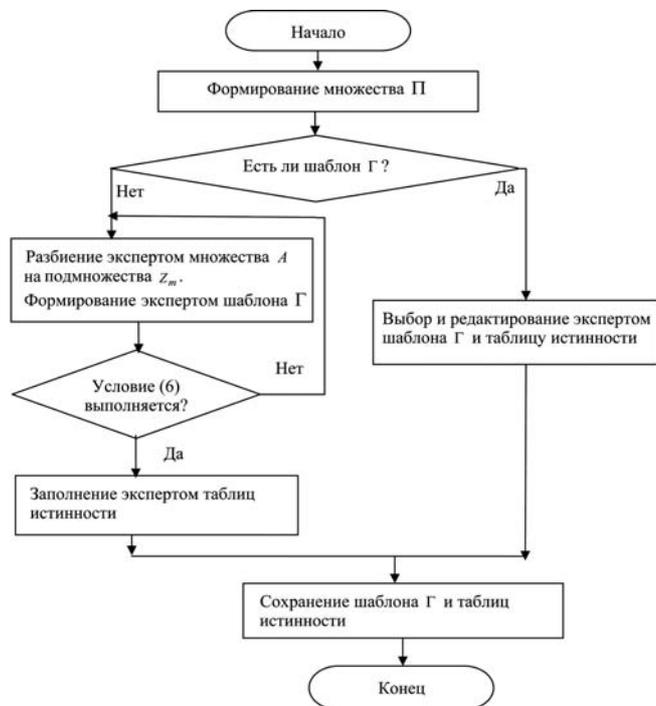


Рис. 3. Процедурная модель заполнения БЗ экспертом

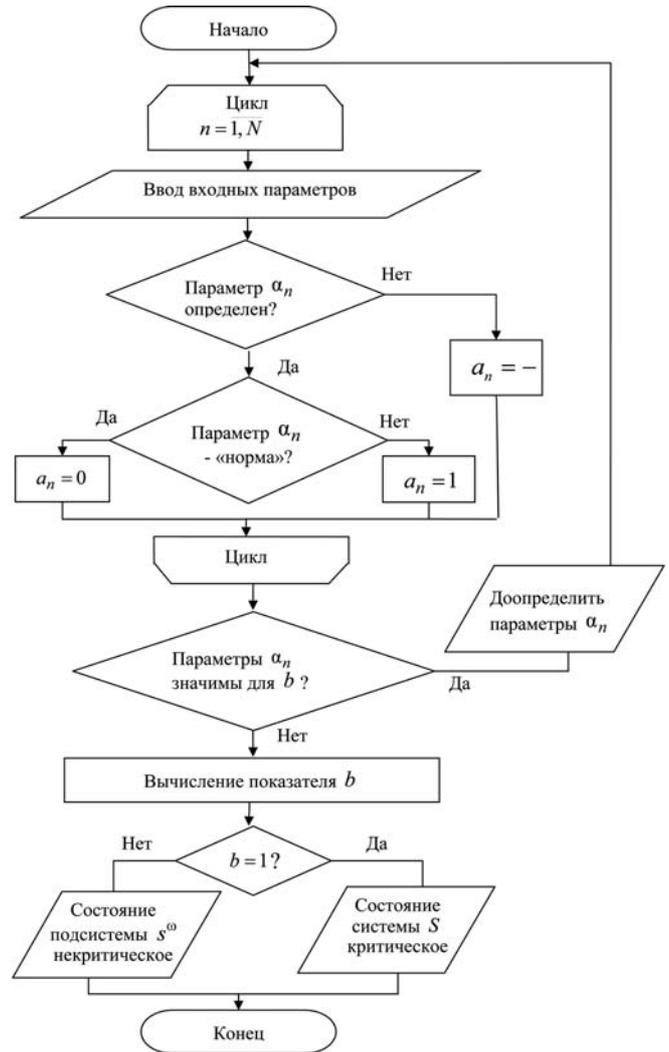


Рис. 4. Процедурная модель определения состояния системы S

при $\beta_n = 2$. Элемент $[\underline{\varepsilon}_n, \overline{\varepsilon}_n]$ есть интервальное число и характеризует норму n -го параметра при $\beta_n = 3$.

БЗ ИСМ представляется в виде информационной модели (3)—(5), (7). Процедурная модель заполнения БЗ показана на рис. 3. Разработана процедурная модель (рис. 4) оценки состояния объекта мониторинга, которая реализуется в ИСМ. Таким образом, информационная и процедурные модели определяют работу ИСМ, которая предназначена для оценки критического состояния объектов, являющихся сложными системами.

Реализация БЗ ИСМ состояния головного мозга человека

Опасным для жизни человека является мозговой инсульт. Комплексный мониторинг состояния головного мозга человека позволяет прогнозировать с высокой степенью достоверности угрозу возник-

новения мозгового инсульта на стадиях, когда еще возможно эффективное использование профилактических мероприятий. Создание для этих целей ИСМ состояния головного мозга человека обеспечит оперативность и объективность полученной информации. БЗ ИСМ состояния головного мозга человека реализуется на основе разработанных информационной (3)—(5), (7) и процедурных моделей (рис. 3, 4).

На основе имеющихся знаний и согласно (7) предлагается нормы входных параметров ИСМ состояния головного мозга человека определять упорядоченным множеством Π в виде последовательности 32 кортежей:

$\Pi = \{ \langle 1 \text{ "международное нормализованное отношение (анализ крови)"}, 3, 0, 0, [0,80, 1,15] \rangle, \langle 2 \text{ "уровень холестерина (анализ крови), ммоль/л"}, 3, 0, 0, [3,9, 5,2] \rangle, \langle 3 \text{ "пульс, удар/мин"}, 3, 0, 0, [60, 80] \rangle, \langle 4 \text{ "систолическое давление, мм рт. ст."}, 3, 0, 0, [120, 140] \rangle, \langle 5 \text{ "диастолическое давление, мм рт. ст."}, 3, 0, 0, [80, 90] \rangle, \langle 6 \text{ "ишемические изменения (электрокардиография — ЭКГ)"}, 2, 0, 0, [0, 0] \rangle, \langle 7 \text{ "рубцовые изменения (ЭКГ)"}, 2, 0, 0, [0, 0] \rangle, \langle 8 \text{ "время анакроты (реоэнцефалография — РЕГ), с"}, 3, 0, 0, [0,1, 0,2] \rangle, \langle 9 \text{ "дикротический индекс (РЕГ), \%"}, 3, 0, 0, [40, 70] \rangle, \langle 10 \text{ "диастолический индекс (РЕГ), \%"}, 3, 0, 0, [50, 75] \rangle, \langle 11 \text{ "частота } \alpha\text{-ритма (электроэнцефалография — ЭЭГ), Гц"}, 3, 0, 0, [8, 13] \rangle, \langle 12 \text{ "амплитуда } \alpha\text{-ритма (ЭЭГ), мкВ"}, 3, 0, 0, [0, 100] \rangle, \langle 13 \text{ "наличие } \alpha\text{-ритма преимущественно в затылочных отделах головного мозга (ЭЭГ)"}, 1, 1, 0, [0, 0] \rangle, \langle 14 \text{ "частота } \beta\text{-ритма (ЭЭГ), Гц"}, 3, 0, 0, [14, 40] \rangle, \langle 15 \text{ "амплитуда } \beta\text{-ритма (ЭЭГ), мкВ"}, 3, 0, 0, [0, 15] \rangle, \langle 16 \text{ "наличие } \beta\text{-ритма преимущественно в лобных отделах головного мозга (ЭЭГ)"}, 1, 1, 0, [0, 0] \rangle, \langle 17 \text{ "частота } \theta\text{-активности (ЭЭГ), Гц"}, 3, 0, 0, [4, 6] \rangle, \langle 18 \text{ "амплитуда } \theta\text{-активности (ЭЭГ), мкВ"}, 3, 0, 0, [0, 40] \rangle, \langle 19 \text{ "наличие } \theta\text{-активности более 15 \% от общего времени регистрации (ЭЭГ)"}, 1, 0, 0, [0, 0] \rangle, \langle 20 \text{ "частота } \delta\text{-активности (ЭЭГ), Гц"}, 3, 0, 0, [0,5, 3] \rangle, \langle 21 \text{ "амплитуда } \delta\text{-активности (ЭЭГ), мкВ"}, 3, 0, 0, [0, 40] \rangle, \langle 22 \text{ "наличие } \delta\text{-активности более 15 \% от общего времени регистрации (ЭЭГ)"}, 1, 0, 0, [0, 0] \rangle, \langle 23 \text{ "диаметр начала левой (мозговой части внутренней сонной артерии) МЧ ВСА (магнитно-резонансная ангиография — МРА), мм"}, 3, 0, 0, [2,439, 2,921] \rangle, \langle 24 \text{ "диаметр конца левой МЧ ВСА (МРА), мм"}, 3, 0, 0, [2,403, 2,877] \rangle, \langle 25 \text{ "длина левой МЧ ВСА (МРА), мм"}, 3, 0, 0, [25,217, 30,183] \rangle, \langle 26 \text{ "диаметр начала правой МЧ ВСА (МРА), мм"}, 3, 0, 0, [2,439, 2,921] \rangle, \langle 27 \text{ "диаметр конца правой МЧ ВСА (МРА), мм"}, 3, 0, 0, [2,403, 2,877] \rangle, \langle 28 \text{ "длина правой МЧ ВСА (МРА), мм"}, 3, 0, 0, [25,217, 30,183] \rangle, \langle 29 \text{ "диаметр начала (базиллярной артерии) БА (МРА), мм"}, 3, 0, 0, [2,529, 3,031] \rangle, \langle 30 \text{ "диаметр конца БА (МРА), мм"}, [2,513, 3,007] \rangle, \langle 31 \text{ "ишемический очаг (магнитно-резонансная томография — МРТ)"}, 2, 0, 0, [0, 0] \rangle, \langle 32 \text{ "ишемическая киста (МРТ)"}, 2, 0, 0, [0, 0] \rangle \}$

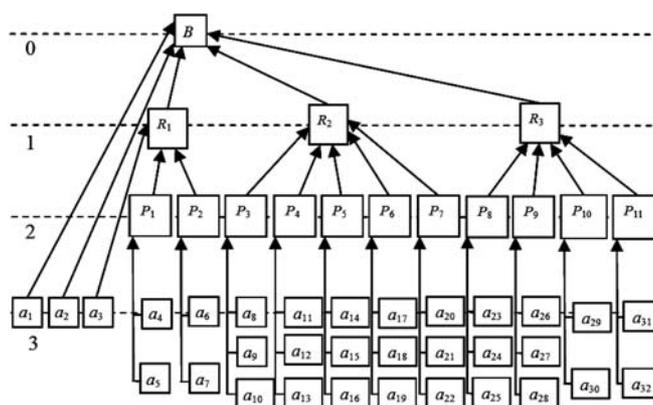


Рис. 5. Иерархическая система блоков показателей БЗ ИСМ состояния головного мозга человека

Первый элемент кортежа определяет название параметра, которое соответствует порядку кортежа во множестве Π . Для первого элемента кортежа записывается номер входного параметра, в скобках ".." указывается соответствующее номеру название и метод исследования, а также единица измерения параметра. На рис. 5 построено дерево иерархии блоков показателей ИСМ состояния головного мозга человека в соответствии с (3)—(6) и на основе декомпозиции булевой функции по методу многоблочной разделительной декомпозиции [5].

На уровне 3 в соответствии с множеством Π представлены входные параметры ИСМ состояния головного мозга человека. Сформированы блоки показателей уровня 2 и уровня 1, которые показаны в таблице. На уровне 0 находится блок B , определяющий значение выходного мониторингового показателя b .

Для каждого блока показателей с участием эксперта построены таблицы истинности, на основе которых в виде совершенной дизъюнктивной нормальной формы [5] получены формулы булевой алгебры (см. таблицу). Здесь норму входного параметра n ($n = \overline{1, 32}$) определяет булева переменная a_n , название которой соответствует номеру кортежа множества Π . Булевы переменные p_l ($l = \overline{1, 11}$), r_k ($k = \overline{1, 3}$) являются соответственно показателями уровней 2 и 3 иерархии блоков показателей, название которых соответствует названию блоков в таблице.

БЗ ИСМ состояния головного мозга человека заполняется экспертом в соответствии с разработанной процедурной моделью (см. рис. 4).

На основе процедурной модели (рис. 5) определяется возможность ($b = 1$) или невозможность ($b = 0$) возникновения мозгового инсульта у объекта мониторинга (пациента). Для этого используются расчетные формулы (таблица), нормы входных параметров в виде множества Π и входные параметры пациента $\{ \langle \alpha_n, \beta_n, \gamma_n, \delta_n^{BX}, \varepsilon_n^{BX} \rangle, n = \overline{1, 32} \}$.

Булевы функции блоков показателей БЗ ИСМ

№	Название блока показателей	Булева функция
Уровень 0 иерархии системы блоков показателей БЗ ИСМ		
1	Возможность или невозможность возникновения инсульта у объекта мониторинга	$b = (a_1 + a_2 + r_1 + r_2 + r_3)(\overline{a_1} + \overline{a_2} + r_1 + r_2 + r_3)(a_1 + a_2 + r_1 + \overline{r_2} + r_3)(a_1 + \overline{a_2} + r_1 + r_2 + r_3)$
Уровень 1 иерархии системы блоков показателей БЗ ИСМ		
2	Кардиологический блок	$r_1 = (a_3 + p_1 + p_2)(\overline{a_3} + p_1 + p_2)(a_3 + \overline{p_1} + p_2)$
3	Блок электрофизиологических исследований головного мозга	$r_2 = (p_3 + p_4 + p_5 + p_6 + p_7)(\overline{p_3} + p_4 + p_5 + p_6 + p_7)(p_3 + \overline{p_4} + p_5 + p_6 + p_7)$ $(\overline{p_3} + \overline{p_4} + p_5 + p_6 + p_7)(p_3 + p_4 + \overline{p_5} + p_6 + p_7)(\overline{p_3} + p_4 + \overline{p_5} + p_6 + p_7)(p_3 + \overline{p_4} + \overline{p_5} + p_6 + p_7)$ $(\overline{p_3} + \overline{p_4} + \overline{p_5} + p_6 + p_7)(p_3 + p_4 + p_5 + \overline{p_6} + p_7)(\overline{p_3} + p_4 + p_5 + \overline{p_6} + p_7)(p_3 + \overline{p_4} + p_5 + \overline{p_6} + p_7)$ $(p_3 + p_4 + p_5 + p_6 + \overline{p_7})(\overline{p_3} + p_4 + p_5 + p_6 + \overline{p_7})(p_3 + \overline{p_4} + p_5 + p_6 + \overline{p_7})(\overline{p_3} + \overline{p_4} + p_5 + p_6 + \overline{p_7})$ $(p_3 + p_4 + \overline{p_5} + p_6 + \overline{p_7})(\overline{p_3} + p_4 + \overline{p_5} + p_6 + \overline{p_7})(p_3 + \overline{p_4} + \overline{p_5} + p_6 + \overline{p_7})(\overline{p_3} + \overline{p_4} + \overline{p_5} + p_6 + \overline{p_7})$
4	Блок морфологического строения головного мозга	$r_3 = (p_8 + p_9 + p_{10} + p_{11})(\overline{p_8} + p_9 + p_{10} + p_{11})(p_8 + \overline{p_9} + p_{10} + p_{11})$
Уровень 2 иерархии системы блоков показателей БЗ ИСМ		
5	Артериальное давление	$p_1 = (a_4 + a_5)(\overline{a_4} + a_5)(a_4 + \overline{a_5})$
6	ЭКГ	$p_2 = (a_6 + a_7)(a_6 + \overline{a_7})$
7	РЭГ	$p_3 = (a_8 + a_9 + a_{10})(\overline{a_8} + a_9 + a_{10})(a_8 + \overline{a_9} + a_{10})(a_8 + a_9 + \overline{a_{10}})(\overline{a_8} + a_9 + \overline{a_{10}})(a_8 + \overline{a_9} + \overline{a_{10}})$
8	α -составляющая ЭЭГ	$p_4 = (a_{11} + a_{12} + a_{13})(\overline{a_{11}} + a_{12} + a_{13})(a_{11} + \overline{a_{12}} + a_{13})(\overline{a_{11}} + \overline{a_{12}} + a_{13})(a_{11} + a_{12} + \overline{a_{13}})$ $(\overline{a_{11}} + a_{12} + \overline{a_{13}})(a_{11} + \overline{a_{12}} + \overline{a_{13}})(\overline{a_{11}} + \overline{a_{12}} + \overline{a_{13}})$
9	β -составляющая ЭЭГ	$p_5 = (a_{14} + a_{15} + a_{16})(\overline{a_{14}} + a_{15} + a_{16})(a_{14} + \overline{a_{15}} + a_{16})(\overline{a_{14}} + \overline{a_{15}} + a_{16})(a_{14} + a_{15} + \overline{a_{16}})$ $(\overline{a_{14}} + a_{15} + \overline{a_{16}})(a_{14} + \overline{a_{15}} + \overline{a_{16}})(\overline{a_{14}} + \overline{a_{15}} + \overline{a_{16}})$
10	θ -составляющая ЭЭГ	$p_6 = (a_{17} + a_{18} + a_{19})(\overline{a_{17}} + a_{18} + a_{19})(a_{17} + \overline{a_{18}} + a_{19})(\overline{a_{17}} + \overline{a_{18}} + a_{19})$
11	δ -составляющая ЭЭГ	$p_7 = (a_{20} + a_{21} + a_{22})(\overline{a_{20}} + a_{21} + a_{22})(a_{20} + \overline{a_{21}} + a_{22})(\overline{a_{20}} + \overline{a_{21}} + a_{22})$
12	Состояние левой МЧ ВСА	$p_8 = (a_{23} + a_{24} + a_{25})(\overline{a_{23}} + a_{24} + a_{25})(a_{23} + a_{24} + \overline{a_{25}})(\overline{a_{23}} + a_{24} + \overline{a_{25}})$
13	Состояние правой МЧ ВСА	$p_9 = (a_{26} + a_{27} + a_{28})(\overline{a_{26}} + a_{27} + a_{28})(a_{26} + a_{27} + \overline{a_{28}})(\overline{a_{26}} + a_{27} + \overline{a_{28}})$
14	Состояние БА	$p_{10} = a_{29} + a_{30}$
15	Исследование МРТ	$p_{11} = (a_{31} + a_{32})(a_{31} + \overline{a_{32}})$

Пациент может иметь неполный набор входных параметров, т. е. $N \neq \emptyset$ и $a_n = -$, при $a_n \in N$. Тогда в зависимости от набора значений других входных параметров $\alpha_n \notin N$ ИСМ предложит доопределить параметры $\alpha_n \in N$ или даст заключение по выходному мониторинговому показателю b .

Предложенный подход к построению ИСМ критических состояний объекта имеет свойства масштабируемости и расширяемости. Например, ИСМ состояния головного мозга человека может дополняться новыми подсистемами, которые обеспечивают мониторинг возможности возникновения других опасных для жизни заболеваний. При этом применяется положение БИП: многократное использование блоков нижнего уровня в других подсистемах [4, 5].

Заключение

Для представления БЗ ИСМ критических состояний объектов удобно использовать аппарат булевой алгебры. Для решения проблемы большой размерности таблиц истинности, с помощью кото-

рых организуются знания, является эффективным применение БИП. Предложенные в работе методы отличаются простотой и быстрой реализацией и могут быть использованы для работы с объектами, представляющими собой сложные системы, такие как административно-территориальные образования, организации, транспортные системы, технологическое оборудование, технологические процессы, экологические и биологические объекты.

Список литературы

1. Подвальный С. Л. Информационно-управляющие системы мониторинга сложных объектов: учеб. пособие. Воронеж: Научная книга, 2010. 164 с.
2. Норенков И. П. Интеллектуальные технологии на базе онтологий // Информационные технологии. 2010. № 1. С. 17–23.
3. Норенков И. П. Основы автоматизированного проектирования. М.: Изд-во МГТУ им. Н. Э. Баумана, 2009. 430 с.
4. Норенков И. П., Уваров М. Ю. Информационно-образовательные среды на базе онтологического подхода // Интернет-порталы: содержание и технологии. Вып. 3. М.: Просвещение, 2005. С. 367–378.
5. Закревский А. Д., Потгосин Ю. В., Черемисинова Л. Д. Логические основы проектирования дискретных устройств. — М.: ФИЗМАТЛИТ, 2007. 592 с.

С. С. Садыков, д-р техн. наук, проф.

e-mail: sadykovss@yandex.ru,

А. С. Белякова, аспирант,

e-mail: asbelyakova@rambler.ru,

ГОУ ВПО Владимирский государственный университет, (филиал) Муромский институт

Математические модели некоторых сердечно-сосудистых заболеваний

Построены математические модели трех сердечно-сосудистых заболеваний. Исследована их работа при изменении параметров функционирования сердца.

Ключевые слова: сердечно-сосудистое заболевание, параметр, корреляция, оценка, регрессионный анализ, значимость, модель, информативность

Введение

Сердечно-сосудистые заболевания (ССЗ) справедливо называют эпидемией XXI века [1]. В течение многих лет они являются основной причиной смертности населения во многих экономически развитых странах, в том числе и в России, где этот показатель составляет 57 % от общей смертности. Согласно статистике последних лет в структуре смертности в нашей стране от ССЗ 85,5 % приходится на долю ишемической болезни сердца (46,8 %) и мозгового инсульта (38,7 %) [2].

При постановке кардиологического диагноза необходимо анализировать около 200 параметров, представленных результатами обследований сердца, индивидуальными характеристиками организма пациента. Поэтому разработка новых способов диагностики заболевания на ранних стадиях, усовершенствование уже существующих с помощью математических методов и информационных технологий является важным и актуальным.

Целью настоящей статьи является построение моделей ССЗ на основе математических методов и информационных технологий обработки данных, а также исследование их работы при изменении параметров функционирования миокарда, выявление наиболее значимых показателей работы сердца.

Построение математических моделей на основе регрессионного анализа данных позволяет выявить зависимость вида заболевания от параметров работы сердца и оценить вклад каждого из показателей в формирование диагноза.

Математическое описание

Особенности регрессионного анализа состоят в следующем [3]. Пусть $x_{i1}, x_{i2}, \dots, x_{in}$ — значения параметров, влияющих на отклик y_i . Связь между каждым значением y_i и $x_{i1}, x_{i2}, \dots, x_{in}$ описывается системой линейных уравнений регрессии

$$\begin{cases} y_1 = a_0 + \sum_{j=1}^n a_j x_{1j}; \\ y_2 = a_0 + \sum_{j=1}^n a_j x_{2j}; \\ \dots \\ y_N = a_0 + \sum_{j=1}^n a_j x_{Nj}; \\ i = 1, \dots, N; j = 1, \dots, n, \end{cases} \quad (1)$$

где a_0 — свободный член; a_1, a_2, \dots, a_n — коэффициенты регрессии — неизвестные параметры, которые необходимо оценить по результатам выборочных наблюдений; N — объем выборки; n — число анализируемых параметров.

Для определения степени взаимовлияния исследуемых показателей рассчитывается корреляционная матрица $R = (r_{x_i x_j})$. Коэффициенты взаимокорреляции $r_{x_i x_j}$ вычисляются известным способом [4].

Коэффициент $r_{x_i x_j}$ является относительной величиной $-1 \leq r_{x_i x_j} \leq 1, i, j = 1, 2, \dots, n$. Сила связи между параметрами оценивается следующим образом:

$$\begin{cases} |r_{x_i x_j}| < 0,3 — слабая; \\ 0,3 \leq |r_{x_i x_j}| \leq 0,7 — умеренная; \\ |r_{x_i x_j}| > 0,7 — сильная. \end{cases} \quad (2)$$

Коэффициенты регрессии $a_0, a_1, a_2, \dots, a_n$ определяются методом наименьших квадратов [4]. Значимость полученных коэффициентов a_1, a_2, \dots, a_n оценивается по t -критерию Стьюдента [2]

$$t_{a_j} = \frac{a_j}{S_{a_j}}, \quad (3)$$

где S_{a_j} — стандартное отклонение коэффициента регрессии a_j .

Полученное значение t_{a_j} сравнивается с таблицей значений t -критерия Стьюдента в зависимости от числа степеней свободы $\nu = n - 1$ [4].

Если полученное для каждого фактора t_j больше табличного t_0 , то коэффициент регрессии a_j считается достоверным при заданной доверительной вероятности $p \leq 0,05$.

Информативность модели проверяется с помощью коэффициента детерминации R_d^2 , который показывает, содержит ли построенная модель достаточную информацию об исследуемом объекте:

$$R_d^2 = 1 - \frac{\sum_{i=1}^N (y_i - f_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (4)$$

где y_i — наблюдаемое значение отклика y ; \bar{y} — среднее арифметическое значение отклика y .

Оценка информативности построенной модели осуществляется следующим образом:

$$\left\{ \begin{array}{l} 0,1 \leq R_d^2 < 0,3 \text{ — слабая;} \\ 0,3 \leq R_d^2 < 0,5 \text{ — умеренная;} \\ 0,5 \leq R_d^2 < 0,7 \text{ — заметная;} \\ 0,7 \leq R_d^2 < 0,9 \text{ — высокая;} \\ 0,9 \leq R_d^2 < 1 \text{ — весьма высокая.} \end{array} \right. \quad (5)$$

С учетом формулы (5) принимается решение об информативности построенной модели заболевания.

Алгоритм построения математических моделей

Алгоритм построения регрессионной модели и ее оценки представлен в виде блок-схемы, приведенной на рис. 1.

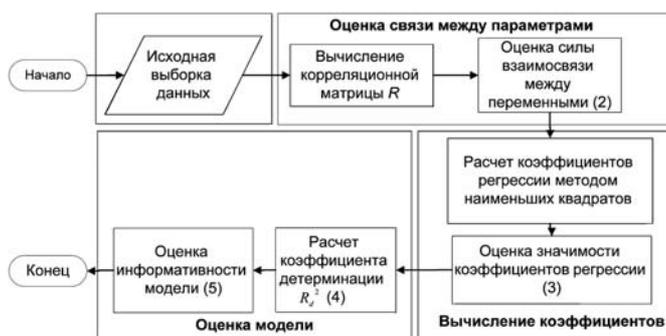


Рис. 1. Блок-схема алгоритма построения моделей сердечно-сосудистых заболеваний

Построение математических моделей трех ССЗ

Рассмотрим построение математических моделей на примере трех ССЗ: полная блокада левой передней ветви пучка Гиса, блокада правой ножки пучка Гиса — часто встречающиеся заболевания; атриовентрикулярная блокада III степени — одно из редких заболеваний [5].

Для построения математических моделей было сформировано три выборки данных № 1, № 2 и № 3 соответственно. Они содержат результаты обследований здоровых пациентов и пациентов, имеющих исследуемые заболевания. Обследование проведено в кардиологическом отделении больницы г. Муром Владимирской области. Возраст обследуемых от 40 до 60 лет. Объем каждой выборки данных составляет 200 результатов обследования пациентов: 100 — здоровые, 100 — имеющие патологию в работе сердца.

В качестве исследуемых используются результаты традиционных кардиологических обследований [5]:

- параметры ЭКГ, полученные с помощью программно-технического комплекса "Миокард 12";
- результаты замера артериального давления;
- индивидуальные характеристики пациента: пол, возраст, индекс массы тела (ИМТ), конституционный тип.

Построение математической регрессионной модели рассматривается на примере обработки данных выборки № 1, содержащей результаты обследований здоровых пациентов и пациентов, имеющих блокаду левой передней ветви пучка Гиса.

Анализируемые параметры обозначаются следующим образом:

- x_{i1} — возраст, полных лет;
- x_{i2} — пол;
- x_{i3} — конституционный тип;
- x_{i4} — ИМТ;
- x_{i5} — частота сердечных сокращений (ЧСС), удар/мин.;
- x_{i6} — угол направления электрической оси сердца (ЭОС), °;
- $x_{i7}, x_{i8}, x_{i9}, x_{i10}$ — параметры ЭКГ: PQ, P, QRS, QT;
- x_{i11} — систолическое артериальное давление (САД), мм рт. ст.;
- x_{i12} — диастолическое артериальное давление (ДАД), мм рт. ст.;
- y_i — диагноз, где i — номер пациента.

Таблица 1

Исходная выборка данных № 1

№ пациента	x_{i1}	x_{i2}	x_{i3}	x_{i4}	x_{i5}	x_{i6}	x_{i7}	x_{i8}	x_{i9}	x_{i10}	x_{i11}	x_{i12}	y_i
1	59	1	4	39	97	-72	0,136	0,09	0,089	0,329	192	77	2
2	49	2	2	15	77	-53	0,128	0,095	0,086	0,367	121	94	2
3	61	2	4	36	62	-96	0,148	0,087	0,134	0,39	139	102	2
...
198	55	1	4	36	78	87	0,124	0,086	0,077	0,346	199	88	1
199	40	1	2	18	86	40	0,140	0,084	0,077	0,393	188	67	1
200	57	1	4	36	87	10	0,116	0,092	0,082	0,361	196	82	1

Значениям параметра x_{j3} даются оценки в баллах [5]: нормостеник — 1 балл, астеник — 2 балла, гиперстеник — 3 балла, ярко выраженное ожирение — 4 балла.

Параметр-отклик y_i принимает два значения: 1 — у здоровых людей, 2 — у пациентов с патологией в работе сердца.

Фрагмент исходной выборки данных № 1 приведен в табл. 1.

Определение силы взаимосвязи факторов

Используя рассчитанные коэффициенты взаимокорреляции между переменными, оценивают силу взаимосвязи по формуле (2). В табл. 2 приведены значения коэффициентов взаимокорреляции, связь между которыми является сильной $|r_{x_i x_j}| > 0,7$, а в табл. 3 — значения коэффициентов корреляции со связями средней силы $0,3 \leq |r_{x_i x_j}| \leq 0,7$.

Таблица 2

Взаимосвязанные переменные с сильными связями, выборка данных № 1

Коэффициент корреляции	$r_{x_3 x_4}$	$r_{x_1 x_{11}}$
Значение	0,85	0,71

Связи между остальными переменными являются выраженными слабо $|r_{x_i x_j}| < 0,3$.

Расчет коэффициентов регрессии

Коэффициенты регрессии определяются согласно методу наименьших квадратов. Вычисленные значения коэффициентов регрессии $a_0, a_1, a_2, \dots, a_n$ приведены в табл. 4. По формуле (3) определяется значение t -критерия Стьюдента для каждого коэффициента регрессии $a_0, a_1, a_2, \dots, a_n$. Полученные значения сравниваются с таблицей t -критерия Стьюдента при уровне значимости $p \leq 0,05$. Если полученное для каждого коэффициента регрессии t_j больше табличного t_0 , то коэффициент регрессии a_j

считается достоверным и принимает участие в уравнении регрессии.

По табл. 4 наиболее значимыми для постановки диагноза "Полная блокада левой передней ветви пучка Гиса" являются параметры x_4 — ИМТ, x_6 — угол ЭОС, x_9 — QRS, x_{10} — QT, x_{11} — САД. Остальные параметры не учитываются при построении результирующего регрессионного уравнения, так как не удовлетворяют условию $p \leq 0,05$. Модель блокады левой ножки пучка Гиса имеет вид

$$y_1 = 0,007x_4 - 0,005x_6 + 3,303x_9 + 1,384x_{10} + 0,001x_{11}. \quad (6)$$

Для определения адекватности модели рассчитывается коэффициент детерминации R_d^2 .

Оценка информативности модели

Для оценки информативности модели блокады левой ножки пучка Гиса (6) по формуле (4) получен коэффициент детерминации $R_d^2 = 0,98$, т. е. информативность модели составляет 98 %. Согласно (5) полученная модель является информативной и может использоваться при диагностике ССЗ.

Аналогичным образом были получены модели остальных исследуемых ССЗ и определена их информативность. Модель блокады правой ножки пучка Гиса выглядит следующим образом:

$$y_2 = -0,049x_2 + 2,203x_7 + 16,020x_9 + 0,001x_{11},$$

$$R_d^2 = 0,96.$$

Модель атриовентрикулярной блокады III степени:

$$y_3 = -0,007x_5 + 2,815x_8 + 4,486x_9 + 0,754x_{10} + 0,001x_{11}, \quad R_d^2 = 0,97.$$

Построение моделей трех ССЗ осуществлено на основе анализа количественных значений параметров, в отличие от работы [6], в которой харак-

Таблица 3

Взаимосвязанные переменные со связями средней силы, выборка данных № 1

Коэффициент корреляции	$r_{x_6 x_3}$	$r_{x_7 x_3}$	$r_{x_9 x_3}$	$r_{x_{10} x_3}$	$r_{x_{11} x_3}$	$r_{x_6 x_3}$	$r_{x_6 x_4}$	$r_{x_7 x_4}$	$r_{x_9 x_4}$	$r_{x_{10} x_4}$
Значение	-0,479	0,392	0,301	0,344	0,344	0,304	-0,543	0,423	0,336	0,419
Коэффициент корреляции	$r_{x_{11} x_4}$	$r_{x_7 x_6}$	$r_{x_9 x_6}$	$r_{x_{10} x_6}$	$r_{x_{11} x_6}$	$r_{x_9 x_7}$	$r_{x_{10} x_7}$	$r_{x_{11} x_7}$	$r_{x_{10} x_9}$	$r_{x_{11} x_{10}}$
Значение	0,339	0,580	-0,572	-0,616	-0,421	0,365	0,453	0,337	0,451	0,350

Таблица 4

Коэффициенты регрессии, выборка № 1

i	0	1	2	3	4	5	6	7	8	9	10	11	12
a_i	-0,010	-0,002	-0,019	-0,001	0,007	0,001	-0,005	0,626	0,596	3,303	1,384	0,001	0,001
t_j	-0,045	-1,295	-0,938	-0,021	2,756	1,365	22,178	1,353	0,801	3,179	4,978	2,028	1,311

теристика ЭКГ выступает как качественный параметр, оцененный в баллах. Естественно, что это отражается на степени информативности модели.

Исследование работы моделей в различных ситуациях

- Проанализируем, какие результаты определения заболеваний будут получены с помощью построенных моделей в различных ситуациях. Для их оценки сформированы три выборки данных. Каждая выборка содержит результаты обследования 50 здоровых пациентов и 50 пациентов, имеющих соответствующее заболевание. Для каждого из пациентов определено значение параметра-отклика у. Определение наличия блокады левой ножки пучка Гиса приведено на рис. 2.

На рисунке видно, что значения параметра-отклика у для здоровых пациентов находятся в интервале (0; 1,2), а для больных — в интервале (1,3; 2). Так как значения параметров работы сердца у каждого пациента имеют отклонения от среднего в пределах нормы, то выходной параметр-отклик у также имеет отклонения.



Рис. 2. Результаты обследования пациентов с блокадой левой передней ветви пучка Гиса и здоровых пациентов



Рис. 3. Определение ССЗ для пациентов с блокадой правой ножки пучка Гиса

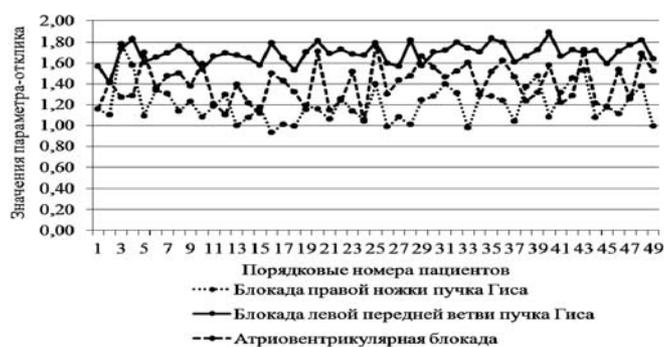


Рис. 4. Поведение моделей для пациентов со всеми заболеваниями

- Рассмотрим поведение полученных математических моделей на примере обработки данных пациентов, имеющих блокаду правой ножки пучка Гиса (рис. 3). Объем выборки 50 записей — результаты обследований пациентов с блокадой правой ножки пучка Гиса.

На рис. 3 видно, что у исследуемой группы пациентов отсутствуют атриовентрикулярная блокада и блокада левой передней ветви пучка Гиса, что свидетельствует о правильном определении заболеваний.

- Рассмотрим определение заболеваний на выборке данных, включающей результаты обследований пациентов со всеми исследуемыми патологиями (рис. 4). Объем выборки — 50 записей.

На рисунке видно, что для всех пациентов, значения параметра-отклика у при диагностике блокады правой ножки и левой передней ветви пучка Гиса и атриовентрикулярной блокады лежат в интервале от (0,8; 2,0). Следовательно, определяется наличие всех трех заболеваний.

Существующие разработки в области построения математических моделей ССЗ представлены небольшим числом работ, использующих в качестве параметров оценки ССЗ результаты биохимических анализов [7], УЗИ [8]. Большая часть из них посвящена выявлению корреляционных взаимосвязей между параметрами без построения аналитической функции, описывающей ССЗ [8, 9].

Построенные модели трех ССЗ основываются на базисных кардиологических обследованиях, проводимых в любой поликлинике. Проведение специальных исследований, требующих дорогостоящего оборудования, не требуется. Таким образом, полученные модели доступны для использования любым врачом-кардиологом при диагностике ССЗ. В качестве результата модель предоставляет одно число, характеризующее наличие данного заболевания либо его отсутствие. Кроме того, полученные модели являются удобным инструментом при проведении статистического анализа заболеваемости.

Результаты построения и исследования математических моделей трех сердечно-сосудистых заболеваний показали следующее:

- наиболее значимыми параметрами при полной блокаде левой передней ветви пучка Гиса являются ИМТ (x_4), угол наклона ЭОС (x_6), значения параметров ЭКГ: QRS (x_9), QT (x_{10}), САД (x_{11});
- у пациентов с высоким ИМТ блокада левой передней ветви пучка Гиса возникает чаще, чем с нормальным ИМТ;
- для блокады правой ножки пучка Гиса большую роль играют такие параметры, как пол пациента (x_2), PQ (x_7), QRS (x_9), значение САД (x_{11});
- у мужчин риск возникновения блокады правой ножки пучка Гиса выше, чем у женщин;
- для атриовентрикулярной блокады III степени ЧСС (x_5), характеристики ЭКГ: P (x_8), QRS (x_9), QT (x_{10}), значение САД (x_{12}) имеют определяющее значение;
- при наличии у человека нескольких ССЗ полученные модели позволяют выявить наличие каждого из заболеваний;
- построенные модели отвечают требованиям информативности и могут быть использованы при диагностике ССЗ.

1. **Садиков С. С., Белякова А. С.** Система поддержки принятия решений при диагностике и оценке состояния сердечно-сосудистой системы. // Интеллектуальные системы: Труды девятого международного симпозиума / Под ред. К. А. Пупкова. М.: РУСАКИ, 2010. С. 488—490.
2. **Российский** статистический ежегодник. М.: Росстат, 2009. 795 с.
3. **Дрейпер Н., Смит Г.** Прикладной регрессионный анализ. М.: Финансы и статистика, 1986.
4. **Афифи А., Эйзен С.** Статистический анализ: подход с использованием ЭВМ / Пер. с англ. М.: Мир, 1982. 488 с.
5. **Национальные** клинические рекомендации. М.: Меди Экспо, 2009. 373 с.
6. **Баевский Р.** Разработка алгоритмов донозологической диагностики на основе применения пошагового регрессионного анализа. Оценка уровня здоровья и риска развития заболеваний с помощью цифрового анализатора биоритмов "Омега-М" // Научные труды съезда физиологов. Т. 2, 2008. С. 45—50.
7. **Бых А. И., Высоцкая Е. В., Порван А. П.** и др. Использование дискриминантного анализа для диагностики хронической сердечной недостаточности у подростков // Вестник НТУ "ХПИ". Сб. научных статей. Харьков: НТУ "ХПИ". 2010. № 31. С. 16—22.
8. **Казаков В. А., Гутор С. С., Суходило И. В.** и др. Корреляционные взаимосвязи гистоморфометрических показателей миокарда различных отделов сердца в норме и при хронической сердечной недостаточности. // Бюллетень сибирской медицины. 2009. № 3. С. 43—44.

8-й МОСКОВСКИЙ МЕЖДУНАРОДНЫЙ ФОРУМ
ТОЧНЫЕ ИЗМЕРЕНИЯ -
ОСНОВА КАЧЕСТВА И БЕЗОПАСНОСТИ

23 – 25 мая '2012
Москва Павильон
ВВЦ №57

СПЕЦИАЛИЗИРОВАННЫЕ
 ВЫСТАВКИ

MetrolExpo

20 мая - Всемирный день метрологии

PromSafety

ResMetering

MedTest

4-й СИМПОЗИУМ
«ТОЧНОСТЬ. КАЧЕСТВО. БЕЗОПАСНОСТЬ»

ДИРЕКЦИЯ ФОРУМА
 Москва, Технопарк ВВЦ, ул. Олонекская, 35, стр. 182,
 Тел/Факс: +7 (495) 937-40-23
 E-mail: metrol@expoprom.ru, www.metrol.expoprom.ru

ОРГАНИЗАТОР
 Федеральное агентство по техническому регулированию и метрологии (Росстандарт)

СОДЕЙСТВИЕ
 Аппарат Правительства Российской Федерации

МЕЖДУНАРОДНЫЕ ПАРТНЕРЫ
 Центр международного промышленного сотрудничества ЮНИДО в РФ
 The International Bureau of Weights and Measures (BIPM)
 International Organization of Legal Metrology (OIML)
 The European Association of National Metrology Institutes (EURAMET)
 The International Committee for Non-Destructive Testing (ICNDT)

С УЧАСТИЕМ
 Минпромторг России, Минэнерго России, Минздравсоцразвития России, Минобороны России, Минобрнауки России, Ростехнадзор, Роскосмос, «Роснано», «Росатом», «Ростехнологии», ОАО «РЖД», ОАО «ОАК», ОАО «Газпром», ОАО «Роснефть», ОАО «ОСК», Российские металлургические группы, ОАО «КАМАЗ», ОАО «АВТОВАЗ», Российский союз химиков, НП «Росхимреактив», Ассоциация аналитических центров «Аналитика», региональные ЦСМ и метрологические институты

ЭКСПЕРТНАЯ КОМИССИЯ
 ФГУ «Ростест -Москва», 32 ГНИИ Минобороны России

ПРОГРАММА ФОРУМА
 8-ая Международная выставка-конкурс средств измерений, испытательного и лабораторного оборудования **«METROLEXPO-2012»**
 Специализированная выставка средств обеспечения безаварийной работы промышленных объектов **«PROMSAFETY-2012»**
 Специализированная выставка коммерческого и технологического учета энергоресурсов **«RESMETERING-2012»**
 2-ая Специализированная выставка СИ для медицинской техники и средств её поверки, контроля и испытаний **«MEDTEST-2012»**
 4-й Московский Международный симпозиум **«ТОЧНОСТЬ. КАЧЕСТВО. БЕЗОПАСНОСТЬ»**
 Всероссийская выставочно-конкурсная программа **«ЗА ЕДИНСТВО ИЗМЕРЕНИЙ»**

УСТРОИТЕЛЬ И ВЫСТАВОЧНЫЙ ОПЕРАТОР
 Компания «Вэстстрой Экспо»

УДК 004

Г. Ковач¹,

В. В. Шилов², канд. техн. наук, зав. каф.,

¹ Компьютерное общество Джона фон Неймана, Венгрия,

² "МАТИ" — РГТУ имени К. Э. Циолковского, Москва

kovacs@mail.datanet.hu, shilov@mati.ru

М-3: к истории компьютера первого поколения

Воспоминания д-ра Гёзы Ковача, одного из разработчиков первой венгерской электронной вычислительной машины М-3, построенной в 1959 г. по переданной из СССР документации, с дополнениями об истории разработки и внедрения этой ЭВМ в разных странах.

Ключевые слова: группа кибернетических исследований АН Венгрии, ЭЦВМ М-3, первый компьютер в Венгрии, первая компьютерная программа в Венгрии, магнитный барабан, первый вычислительный центр в Венгрии

I

Гёза Ковач

М-3: ПЕРВЫЙ ВЕНГЕРСКИЙ ЭЛЕКТРОННЫЙ КОМПЬЮТЕР

Компьютер В-1. Идея построить венгерский компьютер родилась в Центральной пересыльной тюрьме Будапешта. Д-р Режё Тарьян (Rezső Tarján, 1908—1978) в начале пятидесятых был главой совета директоров в Министерстве промышленности. Но Тарьян был также крупным специалистом по математике страхового дела. Он поддерживал тесные профессиональные контакты со многими математиками, в основном с западными. В 1953 г. политическая полиция организовала против него процесс, и он был заключен в тюрьму. Предъявленное обвинение было суровым — шпионаж.

Во время заключения Тарьян работал в техническом подразделении при тюрьме (KÖMI 401)¹ вместе с еще двумя интеллектуалами: физиком Йожефом Хатвани² и инженером-механиком д-ром Ласло Эделени³. Они знали об американских и английских компьютерах и решили построить венгерскую электронную вычислительную машину, подобную EDVAC или EDSAC. Предполагаемый компьютер получил название В(udapest)-1. Они подготовили

эскизный проект разработки, и директор КÖMI 401 послал его в отделение математики Академии наук Венгрии, где его, естественно, отклонили.

В 1955 г. заключенные были освобождены, и Академия разрешила д-ру Тарьяну продолжить начатые в тюрьме исследования. Он сумел организовать компьютерный отдел в недавно созданном Институте счетной техники и приборов Академии наук. Вскоре к нему присоединились несколько молодых инженеров и математиков, а спустя несколько месяцев Академия наук выделила отдел в самостоятельное учреждение, названное Рабочей группой по кибернетическим исследованиям при АН Венгрии (венгерская аббревиатура — МТА ККС).

Директором Рабочей группы был назначен вернувшийся из эмиграции из СССР инженер-электронщик Шандор Варга (Sandor Varga). Тарьян стал его заместителем по науке. Перешли в новый институт и его коллеги из компьютерного отдела — Хатвани и д-р Эделени.

В составе МТА ККС были организованы несколько отделов, которые начали исследования по нескольким направлениям, хотя главной задачей по-прежнему оставалось создание первой венгерской электронно-вычислительной машины — В-1. Тарьян принял на работу в МТА ККС несколько новых сотрудников — математиков, экономистов и инженеров. Математики изучали различные численные методы и методы программирования. Экономисты анализировали возможность решения на ЭВМ тех или иных задач, а также разрабатывали программы для этого.

Кроме того, Тарьян нанял несколько молодых инженеров. *Мне повезло, что я оказался одним из них.* Наша задача состояла в том, чтобы разработать основные электронные узлы будущего компьютера: бистабильные и моностабильные мультивибраторы, вентильные схемы т. д. Все мы были новичками в электронике (дипломы в большинстве своем мы получили в 1956/57 гг.), знали только теорию и не имели никакого опыта практической работы, — вот с таким багажом мы начали разрабатывать устройства В-1.

Это было очень трудно, поскольку, как я сказал, у нас не было практического опыта ни в разработке электронных устройств, ни тем более в создании вычислительных машин. Электронных вычислительных машин никто из нас ранее не видел, хотя первый релейный компьютер (MESz-1), построенный нашим профессором Ласло Козмой⁴ в Будапештском технологическом университете, мы могли изу-

чить. Одним словом, необходимыми для постройки ламповой вычислительной машины знаниями мы не обладали и потому преуспеть не смогли...

В отличие от Тарьяна, Варга длительной разработкой заниматься не хотел. Он считал, что вычислительную машину надо или купить, или, на худой конец, как можно быстрее изготовить хоть что-то годное к употреблению. Варга также видел, что нам — молодым инженерам — недоставало практических навыков для создания своей вычислительной машины. Поэтому первой мыслью Варги было купить электронную вычислительную машину в Советском Союзе, но в 1956 г. советские учреждения компьютеры не продавали. Один из первых советских компьютерных заводов в Пензе начал выпуск машин "Урал" много позже⁵. Так что план Варги был не слишком реалистичным.

В течение 1957 г. Варга и Тарьян несколько раз ездили в Советский Союз. Они посетили, в частности, Энергетический институт⁶. Это была одна из организаций, в которых велась разработка первых советских компьютеров фон-неймановской архитектуры. Эти вычислительные машины были советскими клонами американского компьютера IAS⁷.

Компьютер М-3. Как раз во время визита Варги в Советском Союзе разрабатывалась первая советская малая ЭВМ М-3⁸. Советские коллеги предложили Варге передать свою разработку в МТА ККС, с тем чтобы мы могли построить компьютер в Будапеште.

Напомню, что в то время действовало так называемое "Софийское соглашение", названное так по месту его подписания. Согласно ему страны — члены Совета Экономической Взаимопомощи (СЭВ) должны были бесплатно обмениваться результатами научных разработок⁹. М-3 был результатом научной разработки, поэтому мы могли получить эти документы бесплатно! Варга — и Тарьян с ним согласился — решил воспользоваться этой возможностью, и вскоре — в середине 1957 г.¹⁰ — документация на вычислительную машину прибыла в Будапешт. Насколько я помню, в двух больших ящиках.

Мы знали, что эта разработка была также передана НИИ Кибернетики Эстонии¹¹ и Академии наук Китая. Несколько позже первая ЭВМ М-3 была перевезена из Москвы в Белоруссию, на минский завод им. Орджоникидзе, где было налажено ее серийное производство¹². В Китае также были построены несколько машин М-3, которые стали первыми китайскими ЭВМ¹³. Документация на М-3 была также передана в Армянскую ССР, и на ее основе был сконструирован и построен первый армянский компьютер "Раздан"¹⁴.

Первые построенные в Эстонии¹⁵, Китае, Венгрии и в Советском Союзе машины М-3 основывались на одном и том же прототипе, но все они различались между собой, потому что мы — в наших четырех странах — несколько изменили исходный проект. Кроме того, все коллективы работали не-

зависимо, не имея связи друг с другом. Мы не координировали свои разработки, в результате советская, эстонская, китайская и венгерская ЭВМ М-3 не были совместимы друг с другом. Мы не могли обмениваться программами, — впрочем, в то время мы и не думали, что наши программы нужны кому-либо еще! Каждый писал свои собственные программы и программами коллег вообще не пользовался, а речь о международном обмене программами просто не шла. Важности совместимости программ и обмена программным обеспечением мы не понимали.

В это время Варга реорганизовал технический отдел, снял Тарьяна с должности заместителя по науке и назначил руководителем отдела разработки ЭВМ Балинта Дёмёлки (Balint Dömölki). Я стал его заместителем и отвечал за техническую (электронную) часть работ.

В ходе разработки мои коллеги (и я том числе) предложили много новых решений. Мы изменили некоторые схемы в арифметическом устройстве, ввели новые команды в систему команд, создали новое устройство управления магнитными барабанами (которых было четыре), заменили старые устройства ввода/вывода (телетайп Siemens 100) на более быстрые устройства чтения с перфоленты и записи на перфоленту и т. д.

Через изготовленный мной усилитель я подключил громкоговоритель к моностабильному мульти-

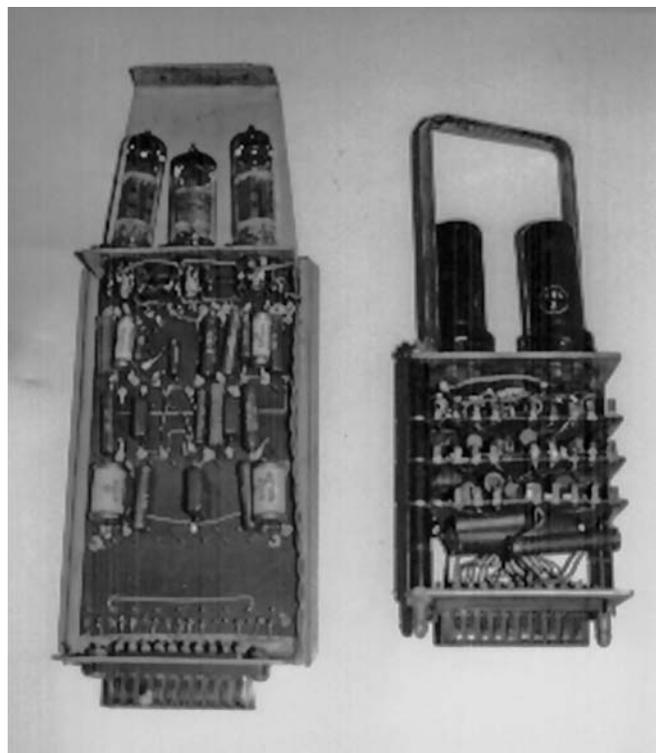


Рис. 1. Ячейки ЭВМ М-3 (Венгрия) на лампах



Рис. 2. Гёза Ковач возле магнитного барабана М-3

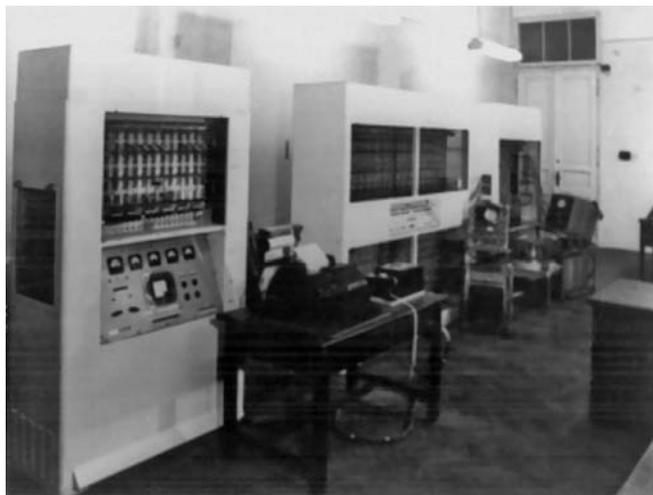


Рис. 3. ЭВМ М-3 (Венгрия)

вибратору в блоке формирования команд. Во время выполнения инструкций мультивибратор генерировал импульсы частотой между 50 Гц и 10 кГц, т. е. в голосовом диапазоне. Так что с начала 1959 г. наш компьютер стал также и "электронным музыкальным инструментом"! Мой коллега написал соответствующую программу, и М-3 исполнял "К Элизе" Бетховена.

В первой версии М-3 мы использовали советские электронные лампы и купроксные диоды, но позже решили заменить их более долговечными лампами венгерского производства (на рис. 1 показаны ламповые ячейки М-3). Сконструированное мною новое устройство управления магнитными барабанами (рис. 2) было полностью изготовлено на новых венгерских лампах "Tungsram"¹⁶. Я также хотел заменить купроксные диоды германиевыми диодами "Tungsram", но, к сожалению, эта попытка оказалась неудачной.

Из Советского Союза мы получили необходимые логические, электронные и прочие схемы, однако никакой рабочей документации на ЭВМ в полученных документах не было. Балинт Дёмёлки начал самостоятельно изучать техническую документацию и постепенно, шаг за шагом разобрался в работе ЭВМ. Он придумал оригинальную концепцию "графической документации", которая объединяла все логические, электронные и другие изображенные графически схемы в единый комплекс. Так что, хотя в конце концов мы смогли построить компьютер, функционирование различных его узлов мы поняли из документации, подготовленной Дёмёлки, а не из полученной документации.

Момент, когда построенный нами компьютер начал работать, был для нас знаменательным. Больше недели мы дни и ночи не покидали институт, никто не уходил домой, — мы чувствовали (были уверены!), что этот момент приближается.

Наши математики подготовили несколько программ, и тоже ожидали, когда их программы заработают. Но когда это произошло, мы никак не хотели поверить: наш компьютер работал!

Его скорость составляла приблизительно 30—50 операций в секунду, и наш М-3 (рис. 3) был на тот момент самым быстрым счетным устройством в стране (а также и самым большим электронным устройством). Поначалу мы даже не поняли, что первая программа проработала успешно, потому что она выполнялась слишком быстро! Видя достигнутые результаты, Варга решил официально передать компьютер представителям Академии наук Венгрии, однако оказалось, что математики — члены Академии — не знали, что такое компьютер... Тогда у Варги возникла превосходная идея: обратиться к одному из лучших советских инженеров, главному конструктору М-3 Г. П. Лопато¹⁷.

21 января 1959 г. было проведено приемочное испытание М-3. Глава комиссии Г. П. Лопато объявил об успешном завершении разработки М-3 и о том, что машина может использоваться не только в институтах Академии наук, но и в университетах, конструкторских бюро, на производстве и т. д. Этого заключения для Президиума венгерской Академии наук было достаточно. Сообщение о завершении разработки компьютера появилось в газетах (рис. 4).

Сразу после успешного приемочного испытания наши математики и экономисты решили на новом компьютере несколько предварительно подготовленных задач, не только научного характера, но и из технической и экономической областей. Многие специалисты: инженеры, экономисты, математики, лингвисты и другие начали изучать программирование, чтобы использовать машину для решения своих задач.



Рис. 4. Страница газеты Esti Hírlap от 21 января 1959 г. с сообщением о разработке первой венгерской ЭВМ

Уже в первые месяцы функционирования на М-3 была решена такая серьезная задача, как вычисления, связанные с разработкой очередного пятилетнего плана развития народного хозяйства. Решались также задачи исследования операций, лингвистического статистического анализа, а также расчета конструкций для строительных работ (например, при восстановлении моста Эржебет в Будапеште) и многие другие.

Первые академические программы. Мы организовали первые в Венгрии курсы по программированию, слушателями которых были математики, инженеры, экономисты и другие исследователи. МТА ККС начал издавать первое компьютерное периодическое издание "Tájékoztató". Наши математики поставили курс лекций на новом факультете программирования в университете.

В 1960 г. наш коллега д-р Бела Креко (Béla Kreko) организовал факультет математического планирования в Университете экономики. Креко хотел, чтобы университет готовил широко образованных экономистов, хорошо разбирающихся в математике и информатике. Я думаю, что такой факультет был одним из первых не только в Венгрии и соседних — т. е. социалистических — странах, но и во всей Европе. Меня пригласили организовать преподавание информатики на этом факультете. Во время обучения студенты могли ознакомиться с компьютером М-3 и с возможностями решения задач на нем. Я написал первый университетский учебник, посвященный компьютерам. В 1965 г. мы — д-р Креко и я — также организовали в этом университете первый в стране университетский вычислительный компьютерный центр — на базе ЭВМ "Урал-2".

Когда в 1959 г. М-3 был успешно испытан и принят венгерской Академией наук, Варга решил — и, естественно, мы поддержали его — что мы должны спроектировать и построить новый, более со-

временный вариант компьютера М-3. Однако он не просил на это санкции Президиума; мы полагали, что разрешения на это тот никогда не даст.

Первый вычислительный центр в Венгрии. Варга еще раз реорганизовал МТА ККС. Он изменил не только название, но и назначение учреждения, создав первый в Венгрии вычислительный центр. В ВЦ Академии наук было несколько отделов; я стал начальником отдела вычислительных работ. Очень скоро мы уже работали в три смены, единственные каникулы у нас были только на Рождество. Каждая смена длилась 8 ч, причем один час занимало техническое обслуживание машины. Электронные лампы работали не слишком надежно, и в каждой смене несколько ламп приходилось заменять.

О том, что мы строим новый, более современный компьютер М-3 без разрешения Президиума Академии наук, стало известно очень скоро. Новый компьютер был готов на 50 %, но Академия велела нам остановить свою "незаконную" работу и демонтировать уже наполовину готовую машину. Они заявили, что на ближайшие пять лет Академии достаточно возможностей уже имеющейся М-3¹⁸. Проступок Варги не остался без последствий. Его уволили из ВЦ, и на его место был назначен начальник отдела экономики д-р Иштван Акзель (Istvan Aczél).

Первый экспорт аппаратных средств из Венгрии. Первая официальная поездка д-ра Акзеля в качестве директора состоялась в 1960 г., когда он посетил конференцию по прикладной математике в Румынии. Там он встретил двух молодых ученых, математика д-ра Йозефа Кауфмана (Josef Kaufmann) и инженера-электронщика Вильгельма Лёвенфельда (Wilhelm Löwenfeld) из Университета Тимишоары. Они рассказали, что построили компьютер, названный ими МЕСИРТ-1, и что память для него они безуспешно пытались купить в Советском Союзе. Они спросили Акзеля, не может ли он помочь. Акзель обратился ко мне с вопросом: "не могли бы мы передать в Тимишоару магнитный барабан?" Я ответил: "Да, можем, потому что в нашем ВЦ имеется несколько запасных барабанов. Если один выходит из строя, мы можем быстро заменить его". Так что мы передали им барабан, а кроме того, чертежи построенного мною ранее устройства управления барабанами. Все, разумеется, бесплатно. Барабан подключили к компьютеру, и он заработал. Это была первая поставка венгерской компьютерной аппаратуры за границу! Спустя какое-то время Академия наук Румынии попросила нас прислать еще три магнитных барабана в Бухарест. Просьбу мы выполнили; больше я об этих барабанах ничего не слышал.

К сожалению, по политическим соображениям МЕСИРТ-1 не был угоден режиму Чаушеску¹⁹. Хотя оба его разработчика были румынскими гражданами, но Кауфман имел венгерское происхождение,

а Лёвенфельд был евреем родом из Германии. Я же — третий проектировщик (барабана) — вообще был иностранцем. МЕСИРТ-1 проработал до 1968 г., после чего был передан музею Банат в Тимишоаре. Чуть позже музей получил указание убрать компьютер из своей экспозиции. С тех пор его судьба длительное время оставалась неизвестной. Я несколько раз бывал в Тимишоаре и пытался найти следы МЕСИРТ-1, но безуспешно.

В 2002 г., в очередной раз читая лекции в Тимишоаре, я встретил молодого журналиста Золтана Патаки (Zoltan Pataki). Я спросил, не знает ли он о судьбе МЕСИРТ-1. Он, в свою очередь, начал наводить справки, и на этот раз нам повезло! Мы обнаружили компьютер в подвале тимшоарской крепости. Телекоммуникационная компания *Alcatel* отремонтировала подвал и экспонировала там компьютер. Всякий раз, бывая в Тимишоаре, я обычно посещал МЕСИРТ-1. Увы, во время последнего визита в конце 2009 г. я узнал, что музей закрыт на реконструкцию, а компьютер был демонтирован и хранится в грязном и неохраняемом помещении... Дух Чаушеску жив!

Передача М-3 в Сегед и его гибель. Вернусь к истории М-3. Наш компьютер работал круглосуточно, на нем решали задачи множество пользователей, причем не только из различных исследовательских институтов и университетов, но и с производства. Они решили много математических, экономических и технических задач; возможность выполнить расчеты на электронной вычислительной машине была в то время для исследователей большой удачей. Я помню множество посетителей из разных стран — СССР, Польши, Румынии, Чехословакии, Франции, желавших увидеть его работу. Несколько раз наш вычислительный центр посещал А. П. Ершов, с которым мы подружились и впоследствии не раз встречались в Москве.

М-3 работал в ВЦ Академии наук до 1965 г., когда в Советском Союзе был куплен новый компьютер "Урал-2" (также ламповый). Компьютер М-3 был передан лаборатории кибернетики университета имени Аттилы Йожефа в Сегеде, которую возглавлял Ласло Кальмар²⁰. Таким образом, мы основали первый в Венгрии вычислительный центр вне столицы. Возглавлял ВЦ сегедского университета д-р Даниэль Мушка (Dániel Muszka).

Спустя три года М-3 признали окончательно устаревшим. 2 января 1968 г. компьютер был остановлен. Его разобрали, а компоненты раздали различным университетским лабораториям.

На мой взгляд, создание М-3 стало событием важнейшего исторического значения. Благодаря ему произошло достаточно раннее знакомство венгерского научного сообщества с компьютерной культурой. М-3 был и остается символом начала компьютерной эры в Венгрии.

II

В. В. Шилов

М-3: КОММЕНТАРИИ И ДОПОЛНЕНИЯ К ИСТОРИИ УНИКАЛЬНОЙ ЭВМ

Известный венгерский ученый, специалист в области компьютерных наук Гёза Ковач родился в 1933 г. В 1957 г. он окончил Будапештский технологический университет с дипломом инженера-электронщика — основные вехи дальнейшей профессиональной деятельности д-ра Ковача упомянуты в его статье. Добавлю лишь, что д-р Ковач является автором первых венгерских университетских учебников по электронике и вычислительной технике, нескольких монографий, а также множества научных и научно-популярных статей в венгерской и зарубежной прессе. Кроме этого, следует отметить его многолетнюю активную деятельность как организатора науки. Так, в 1975—1985 гг. д-р Ковач был Генеральным секретарем, а с 1985 по 1990 и с 1993 по 2000 гг. — вице-президентом венгерского Компьютерного общества Джона фон Неймана; длительное время он представлял Венгрию в одном из технических комитетов (ТС-3) IFIP, организовал и возглавлял в нем рабочую группу по дистанционному обучению; в течение нескольких лет (1996—1998) был советником правительства Венгрии по информатике — и это только небольшая часть его деятельности.

Мне посчастливилось познакомиться с д-ром Гёзой Ковачем в сентябре 2010 г. в Брисбене (Австралия) во время работы Всемирного компьютерного конгресса IFIP (рис. 5). Прочитанный им доклад вызвал огромный интерес у присутствующих, и мне показалось, что для российского читателя воспоминания одного из разработчиков первой венгерской ЭВМ должны оказаться особенно интересными, — тем более, что рассказанная д-ром Ковачем история, насколько мне известно, не отражена ни в одной публикации на русском языке.

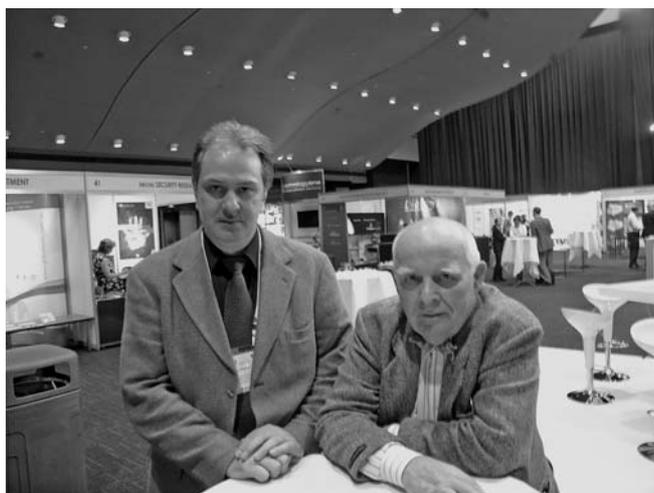


Рис. 5. Валерий Шилов и Гёза Ковач (сентябрь 2010 г.)

Первая часть настоящей статьи является в основном переводом брисбенского доклада д-ра Ковача [4], но учитывает и некоторые другие его публикации, а также доклады на конференции, посвященной 90-летию со дня рождения А. А. Ляпунова (Новосибирск, 2001 г.), конференции MEDICHI 2007 (Methodic and Didactic Challenges of the History of Informatics, Клагенфурт, Австрия, 2007 г.) и др. Д-р Ковач любезно дал свое согласие на их перевод и литературную обработку, а также предоставил в мое распоряжение несколько фотографий.

Кроме того, я считал необходимым снабдить текст воспоминаний д-ра Ковача примечаниями, в которых содержится дополнительная информация об интереснейшей истории разработки советской ЭВМ первого поколения М-3, а также уточняются некоторые утверждения мемуариста. В примечаниях также приводятся сведения о некоторых из упоминаемых д-ром Ковачем венгерских ученых, чьи имена в России недостаточно известны (более подробно об истории информатики и вычислительной техники в Венгрии можно прочитать, например, в [3, 5]). Завершает публикацию таблица, в которой сведены все приводимые в литературе сведения о советских и зарубежных ЭВМ первого поколения, основанных на отечественной разработке М-3.

Примечания

1. Венгерский вариант советских "шарашек".
2. Йожеф Хатвани (József Hatvani, 1926—1987) в 1938—1947 гг. учился в Англии и получил диплом физика в Тринити-колледже в Кембридже. Автор многих пионерских работ в области систем управления и систем с обратной связью. В 1958 г. запатентовал устройство управления станками непосредственно компьютером в режиме реального времени (а не с перфокарты или с магнитной ленты).
3. В конце 1950-х — начале 1960-х гг. Эделени (László Edelenyi) совместно с инженером-электриком Ласло Ладом (Ladó László) сконструировали оригинальную гибридную вычислительную машину EDLA, в которой электронное арифметическое устройство было связано с периферийными устройствами через релейные схемы. Представляет интерес устройство памяти их машины — дисковая память, фактически явившаяся ранним вариантом флоппи-дисков. К 1963 г. им удалось изготовить макет своей машины, но затем разработка была прекращена.
4. С 1930 г. и до Второй мировой войны Ласло Козма (László Kozma, 1902—1983) был одним из ведущих сотрудников отделения компании *Bell* в Антверпене (Бельгия). В 1938 г. он построил свой первый релейный вычислитель: в зависимости от разрядности чисел машина выполняла сложение двух чисел за 1—1,5 с, а умножение за 5—10 с. К концу 1939 г. Козма завершил постройку второй, более сложной машины. Как и первая, она работала в десятичной системе над (максимально) восьми-

разрядными числами. Благодаря наличию встроенных таблиц операция умножения выполнялась всего лишь в два раза медленнее сложения. Машина Козмы обладала таким уникальным по тому времени свойством, как возможность подключения нескольких устройств ввода-вывода (в частности, телетайпов). Кроме того, используемые в вычислениях данные хранились на магнитной ленте. Конструкция релейной машины Козмы была защищена патентами в разных странах, включая Англию и США. После нападения Германии на Бельгию Козма отправил свою машину в США, однако корабль был потоплен немцами. Козма вернулся в Венгрию в 1942 г. После окончания войны был арестован и приговорен к смертной казни, замененной затем тюремным заключением. В 1954 г. был освобожден. В 1955 г. Козма начал конструировать программно-управляемый релейный вычислитель MESz-1, работавший в двоичной системе и завершил эту работу в конце 1958 г. В 1996 г. "за разработку релейных вычислительных машин в 1930-е годы и первых компьютеров в послевоенной Венгрии" посмертно был награжден премией IEEE Computer Pioneer Award.

5. ЭЦВМ "Урал-1" выпускались на Пензенском заводе САМ с 1956 по 1961 гг. Всего было изготовлено 183 экземпляра, которые в условиях плановой экономики в продажу не поступали, а распределялись по предварительным заявкам организаций. ЭЦВМ "Урал-2" была рекомендована к серийному производству в октябре 1959 г. комиссией во главе с академиком А. А. Дородницыным, однако первые пять машин были запущены в производство еще в марте того же года.

6. Речь идет не о Московском энергетическом институте (МЭИ), а об Энергетическом институте (ЭНИИ) АН СССР, в состав которого входила Лаборатория управляющих машин и систем (до 1956 г. — лаборатория электросистем) под руководством И. С. Брука. В 1958 г. на базе лаборатории был организован Институт электронных управляющих машин АН СССР (ИНЭУМ), директором которого стал И. С. Брук.

7. Компьютер IAS, построенный в 1952 г. по проекту Джона фон Неймана в Принстонском Институте перспективных исследований (Institute for Advanced Study, IAS), стал прототипом нескольких ЭВМ первого поколения в США (AVIDAC, ORDVAC, ILLIAC и др.), Австралии (SILLIAC), Швеции (BESK, SMIL) и Израиле (WEIZAC). Однако первые советские ЭВМ создавались совершенно независимо от IAS и общих черт в их архитектуре не слишком много, так что называть их клонами принстонской разработки нет никаких оснований.

8. Разработка малой ЭВМ М-3 (рис. 6) велась под руководством Н. Я. Матюхина, начиная с 1953 г., когда И. С. Брук и директор ВНИИЭМ А. Г. Иосифьян заключили соглашение о проведении совместных работ по созданию ЭВМ для инженерных расчетов. Таким образом, работа по созданию ЭВМ М-3 была

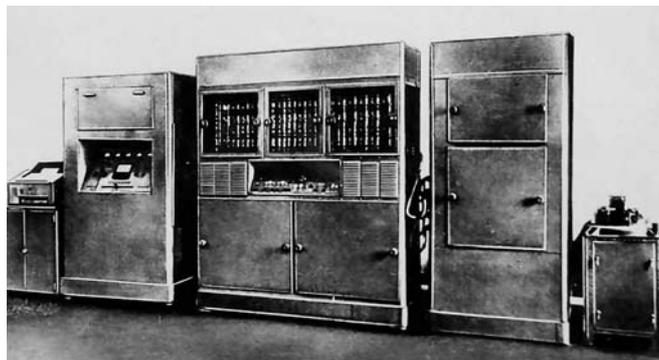


Рис. 6. ЭЦВМ М-3 (СССР)

инициативной (она выполнялась в соответствии с постановлением Президиума АН СССР) и не была включена ни в какие государственные планы — по этой причине проект мог так и остаться невоплощенным "в железе". Ученые и инженеры в то время испытывали острый дефицит вычислительных мощностей, поэтому три организации — КБ С. П. Королева, Всесоюзный научно-исследовательский институт электромеханики (ВНИИЭМ) А. Г. Иосифьяна и Институт математики АН Армянской ССР решили "на паях" изготовить три опытных образца М-3 для своих нужд. ВНИИЭМ взял на себя изготовление и отладку опытных образцов машины, разработку необходимого программного обеспечения и технической документации, пригодной для заводского производства машин, подготовку и проведение государственных испытаний машины и ее продвижение в серийное производство (этими работами со стороны ВНИИЭМ руководил Б. М. Каган). Первый опытный образец был изготовлен в 1956 г., отлажен и в конце того же года предъявлен Государственной комиссии. Согласно воспоминаниям Б. М. Кагана, "Государственная комиссия во главе с академиком Н. Г. Бруевичем с участием М. Р. Шурры-Буры проявила характер и не хотела принимать машину: мол, родилась незаконно. Но все же приняли. И два года не удавалось по-государственному решить вопрос — запустить ее в серийное производство" [2]. Таким образом, ко времени посещения венгерскими инженерами СССР разработка М-3 уже была завершена.

9. СЭВ был образован в 1949 г.

10. В разных публикациях д-р Ковач называет временем прибытия документации середину, конец и осень 1957 г.

11. Институт кибернетики АН Эстонской ССР был организован в 1960 г. По словам д-ра Ковача, информацию о "первом эстонском компьютере" он получил от одного эстонского ученого в 2000 г. Тот же ученый сообщил ему, что "эстонская М-3 проработала долгое время". Однако в известных мне публикациях на русском языке сведений о передаче документации на М-3 в Эстонию нет. Не подтвердил эту информацию и бывший директор НИИ ВК имени М. А. Карцева Юрий Васильевич Рогачев.

В то же время в статье известного эстонского ученого профессора Энна Тыгу говорится, что "в 1960 г. в этом институте {Институте кибернетики АН ЭССР — В. Ш.) был построен первый цифровой компьютер в Эстонии М-3. Оригинальная разработка была выполнена в Минске, но она была существенно улучшена благодаря замене медленного магнитного барабана памятью на ферритовых сердечниках. Это значительно увеличило его производительность" [7]. Таким образом, хотя генетически первая эстонская ЭВМ и восходит к разработке Лаборатории управляющих машин и систем и ВНИИЭМ, все-таки документация на машину пришла в Эстонию не из Москвы, а из Минска. Таким образом, неверно утверждение д-ра Ковача о том, что уже ко времени визита венгерских ученых в Советский Союз "разработка была также передана НИИ Кибернетики Эстонии" — на самом деле это произошло значительно позднее.

12. Д-р Ковач ошибается, говоря о том, что М-3 была "перевезена из Москвы в Белоруссию". В это время как раз завершалось строительство Минского завода счетных машин им. Г. К. Орджоникидзе. Загрузить его было нечем, а поскольку производство машины М-3 не предусматривалось планами ни одного из уже действующих советских заводов-изготовителей, то состоялось решение Госплана СССР об организации серийного производства новой машины в Минске. Как вспоминал Н. Я. Матюхин, "ход событий привел М-3 в Минск, где заканчивалось строительство первого корпуса завода вычислительных машин им. С. Орджоникидзе. Там, в полукустарных условиях, и была выпущена небольшая партия этих машин, вслед за которой завод начал разработку и выпуск широко известной серии машин "Минск" [2]. Действительно, три первые машины М-3 были изготовлены в Минске в 1959 г. В 1960 г. была проведена доработка машины, в частности, магнитный барабан был заменен на память на ферритовых сердечниках. Эта модификация получила название М-3М [1].

13. АН Китая получила техническую документацию на М-3 из Советского Союза в конце 1957 г. и немедленно образовала комиссию по ее изучению. Практически одновременно академия подписала договор с пекинским заводом № 738 о производстве на нем этой машины, и уже в декабре была сформирована рабочая группа из китайских инженеров. Первые работы на заводе начались в марте 1958 г., а к сентябрю был изготовлен работающий макет, получивший название, в переводе с китайского означающее "Мы впервые изготовили вычислительную машину". Позднее его изменили на "1 августа" (в этот день в 1927 г. была образована Народно-освободительная армия Китая). Когда в 1959 г. началось серийное производство вычислительной машины, ее название снова изменили на ЭВМ 103 (рис. 7). Это была почти полная копия М-3, однако китайские инженеры внесли в конст-

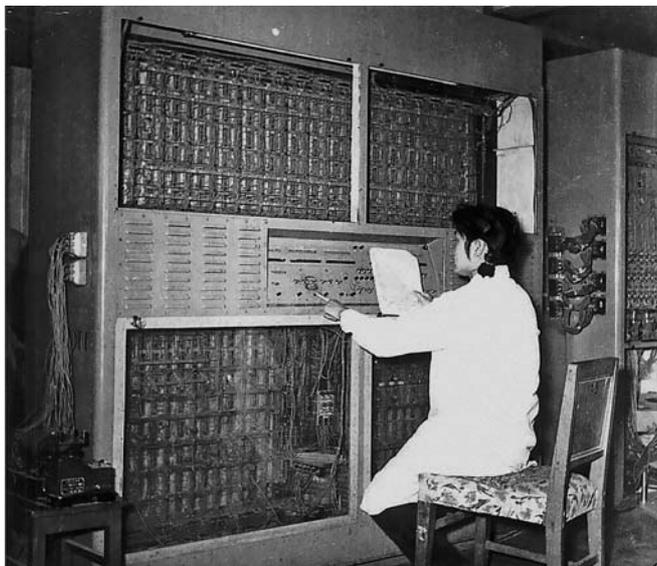


Рис. 7. ЭВМ 103 (Китай)

рукцию некоторые изменения — к магнитному барабану добавили память на ферритовых сердечниках, а также заменили устройства ввода/вывода на более быстрые. В результате производительность ЭВМ и скорость ввода/вывода серьезно увеличились. Интересно, что решение о модификации было принято уже после начала производства, поэтому китайские специалисты разделились на две группы. Одна группа считала, что производство надо приостановить до окончания модификации, вторая — что надо выпускать машину как можно скорее. Решающую роль в принятии окончательного решения в пользу модификации сыграл Г. П. Лопато, который сумел убедить китайцев в том, что после нее машина будет работать более надежно [6].

Внедрение ЭВМ 103 в производство пришлось на начало так называемого "большого скачка", и в погоне за нереальными количественными показателями китайцы не обращали внимания на качество продукции. В результате лишь немногие из выпущенных машин были полностью работоспособны, но и они не отличались высокой надежностью. ЭВМ 103 была передана в военное ведомство и использовалась при решении таких задач, как краткосрочный прогноз погоды, обработка результатов аэрофотосъемки и др.

14. В июне 1956 г. был организован Ереванский научно-исследовательский институт математических машин (ЕрНИИММ); в этом же году одна из трех ЭВМ М-3 в разобранном виде была передана новому институту. По информации, сообщенной мне нынешним директором ЕрНИИММ д-ром техн. наук Саркисом Гургеновичем Саркисяном, работающим в институте с 1958 г., в 1957—1958 гг. коллективом ЕрНИИММ были проведены модернизация, сборка и наладка М-3. Модернизация заключалась во внедрении новой оперативной памяти на ферритовых кольцах объемом в 1024 слова, что позволило

увеличить быстродействие до 3000 оп/с. В 1958 г. после наладки усовершенствованный образец М-3 был передан в Энергетический институт АН СССР для решения задач в области энергетики.

В 1960 г. благодаря накопленному опыту и на основе полученной ЕрНИИММ конструкторской документация на М-3 институт построил свою первую ламповую ЭВМ "Арагац" (разработка велась с 1958 г.; всего были изготовлены 4 экземпляра). Параллельно в это же время по конструкторской документации на М-3 и с технической помощью ВНИИЭМ велась разработка ЭВМ "Раздан". Эта первая в СССР полупроводниковая ЭВМ также была сдана государственной комиссии в 1960 г.

15. См. Прим. 9.

16. Венгерская компания, основанная в 1896 г. и выпускавшая лампочки накаливания и радиолампы. В настоящее время является подразделением General Electric.

17. Главным конструктором М-3 Георгий Павлович Лопато (1924—2003) не был. С 1952 г., после окончания МЭИ, он работал во ВНИИЭМ, участвовал в наладке изготовленной здесь ЭВМ М-3 (см. Прим. 8). В 1958 г. был командирован в Китай для оказания помощи в наладке и запуске в производство китайской ЭВМ 103 (см. Прим. 13). В апреле 1959 г. Г. П. Лопато стал главным инженером СКБ Минского завода счетных машин им. Г. К. Орджоникидзе. Здесь под его руководством как Главного конструктора была разработана и в 1960 г. запущена в серийное производство ламповая ЭЦВМ "Минск-1", прототипом которой также была М-3.

18. Нельзя сказать, что венгерские академики проявили в этом исключительную недалекость. Чуть ранее аналогичные заявления делали не только чиновники, но и крупные ученые и в США, и в Англии, и в СССР (см. В. В. Шилов. "А зачем нужна эта ваша машина?" // Потенциал. 2010. № 6. С. 17—22).

19. В единственной доступной публикации об истории вычислительной техники в Румынии [8] про помощь венгерских коллег не сказано ни слова: разработчиком магнитного барабана для ЭВМ МЕСИРТ-1 в ней назван Василе Балтач (Vasile Baltac). Похоже, что д-р Ковач прав, говоря о политической неужгодности этой ЭВМ политическому руководству Румынии. Не исключено также, что незаинтересованность румынской стороны в этой разработке объяснялась еще и тем, что страна к этому времени уже обладала одним электронным компьютером — построенной в 1957 г. в Институте атомной физики под руководством Виктора Тома (Victor Toma) ЭВМ СИФА-1.

20. Ласло Кальмар (László Kalmár, 1905—1976) — крупный венгерский математик, специалист в области оснований математики и математической логики. Профессор университета в Сегеде, с 1961 г. академик АН Венгрии. "За создание в 1956 г. электронной логической машины" был посмертно удостоен премии IEEE Computer Pioneer Award (1996 г.).

Советские и зарубежные ЭВМ первого поколения, основанные на отечественной разработке М-3

ЭВМ	Год начала выпуска	Разработчик	Выпуск	Элементная база	Разрядность	Память	Производительность	Устройства ввода/вывода	Электропитание
М-3	1956	Лаборатория управляющих машин и систем ЭНИИ им. Г. К. Орджоникидзе, АН СССР, ВНИИЭМ	3	Ламповая. 774 электронные лампы, из них 43 в источниках питания, около 3000 купроксных диодов КВМП-2-7	Двухразрядная 31-разрядная (1 разряд знаковый, 6 — код операции, по 12 разрядов на первый и второй адрес)	Оперативная память на МБ с параллельной выборкой, 2 К слов	30 оп/с	Ввод и вывод данных — стандартная телеграфная аппаратура (трансмисмиттер, телетайп, 7 цифр/с)	От трехфазной сети переменного тока. Потребляемая мощность 10 кВт
М-3	1959	СКБ Минского завода им. Г. К. Орджоникидзе	3 в 1959 г., 16 в 1960 г.	Ламповая. 700 электронных ламп	То же	То же	30 оп/с	—	—
М-3	1960	Институт кибернетики АН ЭССР	1	—	—	—	—	—	—
М-3М	1960	СКБ Минского завода им. Г. К. Орджоникидзе	10 в 1960 г.	Ламповая. 700 электронных ламп	Двухразрядная 31-разрядная (1 разряд знаковый, 6 — код операции, по 12 разрядов на первый и второй адрес)	Ферритовая память 1 К слов (было возможно подключение второго шкафа памяти такого же объема)	1500 оп/с	—	—
Минск-1	1960	СКБ Минского завода им. Г. К. Орджоникидзе	230 (со всеми модификациями)	Электронные лампы, полупроводниковые (диоды)	31 разряд	Ферритовая память 1 К слов, НМП — 65 К слов	Фиксированная и плавающая точка, 3000 оп/с	Ввод с перфоленты, вывод на цифровое печатающее устройство	—
Арагац	1960	ЕрНИИММ	4	Ламповая	—	—	—	—	—
Раздан	1960	ЕрНИИММ	—	Полупроводниковая	Двухразрядная (1 — знак, 5 — порядок, 1 — знак порядка, 29 — мантисса)	—	5000 оп/с	—	От сети переменного тока 220/380 В. Потребляемая мощность 3 кВт
М-3	1959	Рабочая группа по кибернетическим исследованиям при АН Венгрии	1	Ламповая, 1000 вакуумных ламп, 5000 купроксных диодов, 4000 сопротивлений, 3000 конденсаторов	Двухразрядная 31-разрядная (1 разряд знаковый, 6 — код операции, по 12 разрядов на первый и второй адрес)	Сначала: МБ 1 К слов (заем 1,6 К слов). Позднее: внешняя память — два одновременно работающих МБ по 1,6 К слов, основная память на ферритовых сердечниках 1 К слов	Фиксированная точка сложение — 60 мкс, вычитание — 70—120 мкс, умножение — 1,9 мс, деление — 2,0 мс	Сначала: устройство чтения и записи на перфоленту в 5-позиционном телеграфном коде (7 символов/с); телетайп Siemens T-100. Позднее: ввод — фотоэлектрический считыватель с перфоленты Feganti (300 символов/с) в 8-позиционном коде, вывод — перфоратор Credo (100 символов/с)	Потребляемая мощность 10—15 кВт
103	1959	Институт вычислительной техники АН Китая	—	Ламповая	—	МБ и память на ферритовых сердечниках	Фиксированная точка 1800 оп/с	Ввод — фотоэлектрический считыватель с перфоленты (1250 символов/мин), вывод — на печатающее цифровое устройство (650 символов/мин)	—
Примечание. Тире означает, что в литературе явные сведения отсутствуют.									

Список литературы

1. **История** вычислительной техники в Беларуси: Научно-исследовательский институт электронных вычислительных машин / В. Ф. Быченков, Д. Б. Жаворонков, А. М. Жаврид, П. И. Сидорик, Г. Д. Смирнов; под общ. ред. В. Ф. Быченкова, Г. Д. Смирнова. Минск: Вышэйшая школа, 2008. 311 с.
2. **Малиновский Б. Н.** История вычислительной техники в лицах. Киев: фирма "КИТ", ПТОО "А.С.К.", 1995. 384 с.
3. **Brückner H.** Hungarian Pioneers of the Information Age // Hungarian Studies. 1997. Vol. 12. N 1–2. P. 149–167.
4. **Kovács G.** 50 Years Ago We Constructed the First Hungarian Tube Computer, the M-3: Short Stories from the History of the First Hungarian Computer (1957–1960) // History of computing. IFIP World Computer Congress 2010 (WCC-2010). September 20–23, 2010. Brisbane, Australia. Berlin, Heidelberg, New York: Springer, 2010. P. 68–79.
5. **Szentgyörgyi Z.** A Short History of Computing in Hungary // Annals History of Computing. 1999. Vol. 21. N 3. P. 49–57.
6. **Zhang Jiuchun, Zhang Baichun.** Founding of the Chinese Academy of Sciences's Institute of Computing Technology // Annals History of Computing. 2007. Vol. 29. N 1. P. 16–33.
7. **Tyugu Enn.** Computing and Computer Science in the Soviet Baltic Region // History of Nordic Computing 2. Second IFIP WG 9.7 Conference on the History of Nordic Computing (HiNC2), August 21–23, 2007, Turku, Finland. Revised Selected Papers. Ed. by J. Impagliazzo, T. Järvi, P. Paju. Berlin, Heidelberg, New-York: Springer, 2009. P. 29–37.
8. **Anonymous.** History of Computer Developments in Romania // AHC. July–September 1999. Vol. 21. N 3. P. 58–60.

Указатель статей, опубликованных в журнале "Информационные технологии" в 2011 г.

ОБЩИЕ ВОПРОСЫ ИНФОРМАТИЗАЦИИ

- Гуревич И. М.** Атомы, молекулы и фундаментальные ограничения на информационные характеристики систем. № 9.
- Гюльмамедов Р. Г.** Определение критических проблем информатизации регионов. № 9.

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, СЕТИ И СИСТЕМЫ СВЯЗИ

- Амиршахи Б.** Кластеризация GRID-ресурсов для оптимизации информационного обмена при совместной обработке результатов распределенных вычислений. № 2.
- Андреев Д. В.** Аналоговые мультиплексоры для логической обработки континуальных данных. № 10.
- Апарина Е. Ю., Бегаев А. Н., Куделя В. Н.** Проблемы и решения по доставке информации приложений реального времени в IP-сетях. № 7.
- Асадов Х. Г., Исмаилов К. Х.** Информационный метод позиционирования объектов. № 9.
- Бобков С. Г.** Архитектурные и конструктивные решения вычислительных систем с производительностью 1–100 Тфлопс. № 10.
- Богоявленский Ю. А., Кулаков К. А., Корзун Д. Ж.** Линейные диофантовы модели восстановления соединений в сетях MPLS. № 3.
- Гливенко Е. В., Прядко С. А., Фомочкина А. С., Левонян К. А.** Методы решения систем линейных и нелинейных алгебраических уравнений, обладающие естественным параллелизмом. № 10.

- Журавлева Л. М.** Закон Мура и развитие устройств обработки и передачи информации. № 11.
- Ибрагимов Б. Г., Ибрагимов Г. Г.** Эффективность передачи мультимедийного трафика многофункциональным абонентским и сетевым терминалом. № 9.
- Касумова Р. Т.** Сравнительный анализ географических доменов верхнего уровня сети Интернет. № 7.
- Львович Я. Е., Львович И. Я., Преображенский А. П., Головинов С. О.** Исследование метода трассировки лучей при проектировании беспроводных систем связи. № 8.
- Мамченко А. Е., Першеев В. Г.** О кодах представления чисел с фиксированной точкой (запятой) в компьютерах и вычислительных системах. № 4.
- Мансуров Т. М., Мамедов И. А., Мансуров Э. Т.** Разработка методики определения длины регенерационного участка xDSL-модемов сети абонентского доступа. № 2.
- Мошевикин А. П., Галов А. С., Волков А. С.** Локация в беспроводных сетях датчиков стандарта nanoLOC (IEEE 802. 15. 4a). № 8.
- Наумова В. В., Горячев И. Н.** Разработка системы видеоконференцсвязи отделения наук о Земле РАН. № 3.
- Перепелкин Д. А., Перепелкин А. И.** Алгоритм адаптивной ускоренной маршрутизации в условиях динамически изменяющихся нагрузок на линиях связи в корпоративной сети. № 3.
- Саак А. Э.** Алгоритмы диспетчеризации в Grid-системах на основе квадратичной типизации массивов заявок. № 11.
- Саак А. Э.** Локально-оптимальные ресурсные распределения. № 2.
- Сериков Д. А.** Применение механизмов контроля насыщения для разделения ресурсов в распределенной вычислительной среде. № 3.
- Тельпухов Д. В.** Построение обратных преобразователей модулярной логарифметики для устройств цифровой обработки сигналов. № 4.

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Асадов Х. Г., Алиева Е. Н.** О возможности применения метода многоуровневой оптимизации для оптимальной информационной оценки разнообразия видов в экологических системах. № 8.
- Ахи А. А., Станкевич А. С., Шалыто А. А.** Алгоритм построения флибов со 100%-ной точностью предсказания. № 7.
- Бояринов Ю. Г., Борисов В. В.** Анализ систем и процессов на основе нечетких полумарковских моделей. № 11.
- Бронштейн Е. М., Заико Т. А.** Задача маршрутизации с запретами. № 6.
- Бунькина Н. И., Лакеев Д. В.** Задачи составления расписаний с альтернативными приборами. Решение с помощью генетических алгоритмов. № 9.
- Владимирский Э. И., Исмаилов Б. И.** Нелинейный рекуррентный анализ как математическая модель управления хаотическими процессами. № 5.
- Гильманова Н. А.** Мультиметодный алгоритм трехмерной упаковки с регулируемым провисанием объектов в контейнере. № 8.
- Елтаренко Е. А.** Аппроксимация предпочтений в многокритериальных задачах операторами агрегирования. № 6.
- Елтаренко Е. А.** Операторы мягкой логики в многокритериальных задачах. № 10.
- Казаков П. В.** Генетические алгоритмы многокритериальной оптимизации. Обзор. № 10.
- Квасов И. Е., Петров И. Б., Санников А. В., Фаворская А. В.** Компьютерное моделирование пространственных динамических процессов сеточно-характеристическим методом на неструктурированных тетраэдральных сетках. № 9.
- Ковалевич А. А., Якимов А. И., Албкеират Д. М.** Исследование стохастических алгоритмов оптимизации для применения в имитационном моделировании систем. № 8.
- Кузьмина И. А.** Генетические методы синтеза сетей энергоснабжения. № 11.
- Кухаренко Б. Г., Пономарев Д. И.** Нелинейная байесовская фильтрация многомерных временных рядов. № 6.
- Максименко-Шейко К. В., Толок А. В., Шейко Т. И.** R-функции в фрактальной геометрии. № 7.
- Мокшин В. В., Якимов И. М.** Метод формирования модели анализа сложной системы. № 5.
- Норенков И. П.** Редукция графа работ в задаче синтеза расписаний. № 10.
- Орехов Э. Ю., Орехов Ю. В.** Об оценке качества эвристического алгоритма на конечной массовой задаче. № 7.
- Пашенко В. Н., Мартынюк В. А., Моников А. С., Пашенко В. В.** Исследование влияния типа конечных элементов на продолжительность и точность расчета температурных напряжений в среде ANSYSWorkbench. № 8.

- Петров И. Б., Фаворская А. В.** Библиотека методов интерполяции высоких порядков на неструктурированных треугольных и тетраэдральных сетках. № 9.
- Прилуцкий М. Х., Власов В. С.** Построение оптимальных по быстродействию расписаний в канонических системах "конвейер—сеть". № 3.
- Рудаков И. В., Ребриков А. В.** Неполная верификация сложных дискретных систем. № 3.
- Семин В. В.** Алгоритмы построения и оптимизации симплицальных комплексов на квазистационарных решетках. № 5.
- Семин В. В.** Моделирование поведения стохастической триангуляции на квазистационарных решетках. № 6.
- Стрижов В. В., Крымова Е. А.** Выбор моделей в линейном регрессионном анализе. № 10.
- Филиппова А. С., Фролов Р. В.** О создании оперативных сетевых графиков работ бригад текущего и капитального ремонта скважин. № 6.

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

- Болховитянов А. В., Гусев С. В., Чеповский А. М.** Модель и база знаний глагольного управления в предложениях на русском языке. № 12.
- Болховитянов А. В., Чеповский А. М.** Методы автоматического анализа словоформ. № 4.
- Бородащенко А. Ю., Рябцев А. О.** Алгоритм оценки массива гипертекстовых документов на семантическое сходство с эталоном. № 6.
- Бородащенко А. Ю., Яковлев В. А.** Алгоритм фильтрации текстовой информации на основе марковской модели. № 5.
- Дербенев Н. В., Толчеев В. О.** Выявление нечетких дубликатов в наукометрическом анализе. № 12.
- Домрачев В. Г., Полещук О. М.** Определение рейтинговых оценок совокупностей объектов и мониторинг их состояний в условиях нечеткой исходной информации. № 6.
- Евгеньев Г. Б.** Синергетическая методология интеграции знаний. № 1.
- Еременко Ю. И., Глущенко А. И.** О решении неформализуемых и плохоформализуемых задач методами иммунных алгоритмов. № 7.
- Карпенко А. П.** Методика оценки релевантности документов онтологической базы знаний. № 4.
- Колесникова С. И.** Особенности применения линейной свертки критериев в методе парных сравнений. № 1.
- Комаров Е. Г.** Построение модели эталонного объекта и рейтинговых оценок на его основе. № 1.
- Леденева Т. М., Нгуен Нгок Хуи.** Влияние функции подобия на результаты нечеткой классификации. № 11.
- Масликов В. И.** Универсальная система категорических суждений силлогистики. № 8.
- Мочалова А. В., Мочалов В. А.** Интеллектуальная вопросно-ответная система. № 5.

- Назаров Д. А.** Задача двухмерной упаковки в полосу: точный алгоритм с разбиением на лестничные частичные упаковки. № 5.
- Норенков И. П.** Документальные базы знаний на основе онтологий. № 2.
- Савченко А. В.** Трехпороговая система для распознавания изображений на основе метода направленного перебора. № 11.
- Садыков С. С., Савичева С. В.** Идентификация реальных плоских объектов на основе единственного признака точек их внешних контуров. № 8.
- Сафронов В. В.** Сравнительная оценка методов "жесткого" ранжирования и анализа иерархий в задаче гипервекторного ранжирования систем. № 7.
- Толчеев В. О.** Анализ проблемы и разработка процедуры выявления нечетких дубликатов научных статей по библиографическим описаниям. № 2.
- Трофимов А. Г., Скругин В. И.** Системы нейрокомпьютерного интерфейса. Обзор. № 2.
- Ходашинский И. А., Синьков Д. С.** Идентификация параметров нечетких систем на основе адаптивного алгоритма роящихся частиц. № 8.
- Юсупова Н. И., Валеев Р. С.** Рациональное размещение грузов в контейнеры с учетом их физических характеристик с помощью роботизированного комплекса. № 12.
- Якимов В. Н., Мошков И. С.** Методика анализа сложных составных терминов в тексте на естественном языке. № 11.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

- Алымова Е. В.** Критерий полноты тестовых наборов, ориентированных на проверку распараллеливающих преобразований программ. № 9.
- Беневоленский С. Б., Кириллов Ю. И., Пожар В. Э., Пустовойт В. И., Спиридонов И. С.** Разработка платформы программного комплекса для акустооптических спектрометров. № 6.
- Бобков В. А., Мельман С. В.** Параллельная трассировка октантных деревьев на языке CUDA. № 4.
- Бобченков А. В., Топорков В. В.** Метод оптимального планирования управляемых потоков заданий в распределенных вычислительных средах. № 5.
- Брусенцов Л. Е.** Архитектура инструмента поиска оптимальных ключей компиляции. № 9.
- Зуев А. С., Кучеров О. Б.** Внедрение контекстно-зависимых элементов управления в графические интерфейсы компьютерных программ. № 5.
- Лидовский В. В.** Анализ LR-разбора для 18 языков программирования. № 12.
- Пекунов В. В.** Метаслой моделирования алгоритма, данных и функциональных характеристик последовательных и параллельных программ. № 6.
- Полуян С. В.** Уточнение графа информационных связей с помощью анализа псевдонимов. № 4.

- Рогозов Ю. И., Свиридов А. С., Дегтярев А. А.** Анализ и перспективы развития разработки программного обеспечения. № 12.
- Таганов А. И.** Методы идентификации, анализа и сокращения проектных рисков качества программных изделий в условиях нечеткости. № 9.
- Шейпак С. А., Шилов В. В.** Архитектура распределенного сервиса создания цифровой графики и анимации для интернет-систем. № 5.

СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

- Бибило П. Н., Логинова И. П., Романов В. И., Черемисинова Л. Д.** Система CLTT проектирования топологии функциональных блоков заказных цифровых СБИС. № 1.
- Гридин В. Н., Дмитриевич Г. Д., Анисимов Д. А.** Построение систем автоматизированного проектирования на основе Web-технологий. № 5.
- Матюшкин И. В.** Перспективы развития современных средств проектирования клеточных автоматов. № 4.
- Талалай М. С., Трушин К. В.** Обзор подходов к проектированию регулярных ИС. № 1.
- Талалай М. С., Трушин К. В., Венгер О. В.** Логический синтез комбинационных схем на основе транзисторных шаблонов с регулярной топологией. № 4.
- Черемисинов Д. И., Черемисинова Л. Д.** Минимизация двухуровневых КМОП-схем с учетом энергопотребления. № 5.

ГЕОИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Архипова О. Е., Бойко В. В., Ковалева Г. В., Москаленко В. А., Тарасова Т. Т.** Методические подходы к оценке природных и антропогенных изменений Азово-Черноморского побережья. № 11.
- Беляков С. Л., Белякова М. Л., Самойлов Д. С.** Геоинформационный сервис ситуационного центра. № 8.
- Герасимов Ю. Ю., Соколов А. П., Катаров В. К.** Разработка системы оптимального проектирования сети лесовозных автомобильных дорог. № 1.
- Ефанов Д. В., Ибрагимов А. М.** Динамические условные знаки и способы их реализации в современных геоинформационных системах. № 11.
- Замятин А. В.** Концепция региональной информационной системы аэрокосмического мониторинга с интеллектуальной распределенно-параллельной обработкой данных. № 7.
- Кобзаренко Д. Н.** Особенности применения некоторых алгоритмов вычислительной геометрии в геоинформационных технологиях. № 8.
- Мандрикова О. В., Соловьев И. С.** Вейвлет-технология анализа вариаций магнитного поля Земли. № 1.

- Струченков В. И., Козлов А. Н., Егунов А. С.** Динамическое программирование в проектировании трасс линейных сооружений. № 8.
- Струченков В. И., Козлов А. Н., Егунов А. С.** Кусочно-параболическая аппроксимация плоских кривых при наличии ограничений специального вида. № 7.

БЕЗОПАСНОСТЬ ИНФОРМАЦИИ И УПРАВЛЕНИЕ РИСКАМИ

- Алгулиев Р. М., Назирова С. А.** Об одном подходе к формированию и реализации политики борьбы со спамом с учетом требований прав человека. № 4.
- Берников А. Р., Графов Р. П., Лысенко С. Н., Савенко О. С.** Поиск вредоносных программ в распределенных тренажерах с использованием технологии нечеткой логики. № 10.
- Борисов В. В., Сеньков А. В.** Интеллектуальное управление рисками в сложных организационно-технических системах. № 10.
- Дрюченко М. А., Сирота А. А.** Нейросетевые модели и алгоритмы стеганографического скрывания информации. № 3.
- Жуков И. Ю., Михайлов Д. М., Стариковский А. В.** Усовершенствованный протокол аутентификации бюджетных RFID-меток. № 7.
- Имамвердиев Я. Н., Деракшанде С. А.** Сервис-ориентированная эталонная модель для управления рисками информационной безопасности. № 3.
- Коробицын В. В., Ильин С. С.** Реализация симметричного шифрования по алгоритму ГОСТ-28147 на графическом процессоре с использованием технологии CUDA. № 4.
- Червяков Н. И., Бабенко М. Г.** Пороговая схема разделения секрета на эллиптической кривой. № 2.
- Чистякова Т. Б., Садыков И. А., Колерт К., Иванов А. Б.** Методы кодирования и идентификации упаковок фармацевтической продукции для защиты от фальсификации. № 7.

WEB-ТЕХНОЛОГИИ

- Васенин В. А., Шундеев А. С.** Развитие веб-технологий и промежуточного программного обеспечения. № 12.
- Тарнавский Г. А., Чесноков С. С.** Анализ клиентской базы Центра компьютерного моделирования, "облачного" портала Интернета. № 12.

РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ

- Амиршахи Б.** GRID-технологии решения больших систем линейных уравнений на вычислительной сети и на суперкомпьютере кластерного типа. № 6.
- Топорков В. В., Целищев А. С.** Метод критических работ как перспектива эффективной организации распределенных вычислений. № 6.

КОМПЬЮТЕРНАЯ ГРАФИКА

- Архипов О. П., Зыкова З. П.** Многокритериальный выбор тестового множества при исследовании цветовосприятия. № 2.
- Бобков В. А., Роньшин Ю. И.** Реализация алгоритма построения карт глубин на графических процессорах. № 8.
- Гулаков В. К., Трубаков А. О.** Эффективный алгоритм преобразования полноцветного изображения к палитре для систем поиска по содержанию. № 8.
- Левашкина А. О., Поршнев С. В.** Исследование возможности использования ключевых точек в задаче поиска изображений с визуально похожими объектами. № 2.

БАЗЫ ДАННЫХ

- Ильясов Б. Г., Левков А. А.** Структурная оптимизация реляционных моделей сложных иерархических систем. № 3.
- Рогозов Ю. И., Свиридов А. С., Кучеров С. А.** Метод построения структурно-независимых баз данных с использованием реляционных технологий. № 2.
- Рубцов Е. А.** Модель данных для сбора, хранения и обработки информации о существующих во времени объектах различных классов. № 2.
- Халабия Р. Ф.** Организация и структура динамических распределенных баз данных. № 3.
- Шкляев Д. А.** Формальная верификация понятий отказоустойчивости для распределенных баз данных. № 2.

ИНФОРМАЦИОННО-ИЗМЕРИТЕЛЬНЫЕ И ДИАГНОСТИЧЕСКИЕ СИСТЕМЫ

- Иванов И. А., Увайсов С. У.** Информационная модель процесса проектирования контролепригодных радиоэлектронных средств. № 12.
- Кузнецов Л. А.** Меры оценки состояния объектов и процессов при мониторинге. № 12.
- Малыхина Г. Ф.** Статистическая модель для группы параметров, контролируемых в интеллектуальной измерительной системе. № 11.
- Сперанский Д. В.** Идентификация состояний одного типа билинейных систем. № 12.

ОБРАБОТКА СИГНАЛОВ

- Замятин А. В.** Дифференцированное сжатие аэрокосмических изображений с потерями. № 6.
- Трофимов А. Г., Скругин В. И.** Метод выделения динамических паттернов в задаче классификации многомерных временных рядов. № 4.
- Цыцулин А. К., Фахми Ш. С., Колесников Е. И., Очкур С. В.** Функционал взаимобмена сложности и точности систем кодирования непрерывного сигнала. № 4.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Джашитов В. Э., Панкратов В. М., Голиков А. В.** Мультимедийные информационно-компьютерные технологии в научно-образовательных курсах лекций и экспресс-контроле знаний по точным наукам. № 1.
- Зимин А. М., Шумов А. В., Кривицкий С. Е., Троинов В. И.** Автоматизированная сетевая учебно-научная лаборатория по спектроскопии плазмы. № 6.
- Пузанков Д. В., Пантелеев М. Г., Жукова Н. А., Семенов В. С.** Информационная среда проектирования компетентностных моделей на основе технологий семантического Web. № 1.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И УПРАВЛЕНИИ

- Зак Ю. А.** Система вероятностных критериев и ограничений в задачах выбора оптимального портфеля инвестиций. № 10.
- Куприянов Ю. В., Таратухин В. В.** Принципы управления выгодами на проектах внедрения корпоративных информационных систем. № 6.
- Савченко В. В.** Информационная теория колебаний биржевых котировок в динамике. № 3.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В МЕДИЦИНЕ

- Васюков А. В., Петров И. Б.** Компьютерное моделирование последствий механических черепно-мозговых травм. № 5.
- Галкин В. А., Белый Ю. А., Кучеров А. А.** Информационное и программное обеспечение антимикробной фотодинамической терапии. № 8.
- Кузнецов А. А.** Принципы структурно-топологического анализа диаграмм ритма сердца. № 9.
- Кузнецов А. А.** Системная связь между процессами ритма сердца и динамики систологического потенциала. № 8.
- Садыков С. С., Белякова А. С.** Математические модели некоторых сердечно-сосудистых заболеваний. № 12.
- Сирота Е. А.** Информационная система управления производством гемоконпонентов "Службы крови". № 9.
- Фролов С. В., Горбунов А. В., Лядов М. А., Фареа С. Г.** Блочный-иерархический подход для построения базы знаний информационной системы мониторинга состояния головного мозга человека. № 12.

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Беневоленский С. Б., Кириллов Ю. И., Колосянко Д. М., Пожар В. Э., Пожар К. В., Пустовойт В. И.** Совершенствование программного обеспечения для обработки данных, получаемых с акустооптических спектрометров. № 10.
- Коложнов В. В., Колотов В. В., Сединин В. И.** Новый подход к распознаванию номерных знаков и оценка влияния различных факторов на эффективность распознавания. № 7.
- Малафеев С. С., Черняев А. В.** Информационная система визуализации работы экскаватора. № 10.
- Масюк М. А.** Система анализа и визуализации связей нормативно-правовых документов. № 2.
- Трахтенгерц Э. А.** Информационные технологии коррекции целей и стратегических решений в динамике манипулирования общественным мнением. № 10.

ПРОГРАММНЫЕ ПРОДУКТЫ И СИСТЕМЫ

- Морылев Р. И.** Инструмент определения распараллеливаемых циклов. № 10.
- Прохоров А. И., Туманов В. Е., Психа Б. Л., Поликарпов К. И.** Информационно-аналитическая система учета результатов интеллектуальной деятельности в научной организации. № 10.

ИСТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

- Ковач Г., Шилов В. В.** М-3: к истории компьютера первого поколения. № 12.

Журнал в журнале

НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ

- Аведьян Э. Д., Галушкин А. И., Пантюхин Д. В.** Ассоциативная нейронная сеть СМАС и ее модификации в задаче распознавания образов. № 7.
- Аведьян Э. Д., Пантюхин Д. В.** Алгоритм нелинейного преобразования аргументов в нейронной сети СМАС. № 1.
- Вичугов В. Н.** Алгоритм настройки радиально-базисной нейронной сети. № 7.
- Воронков И. М., Кречетов И. В., Харламов А. А.** Обработка больших массивов текстовой информации и перспективы ее развития для информационно-аналитических систем, программная и аппаратная реализация. № 3.
- Галушкин А. И.** Аналитические методы и нейросетевые технологии в решении задач по программе "Протеом человека". № 3.
- Галушкин А. И.** Нейрокомпьютеры в решении задач обеспечения информационной безопасности. № 1.
- Галушкин А. И.** Нейросетевые технологии в решении задач нейрореставрологии. № 9.

Галушкин А. И., Казанцев П. А. Нейросетевое распознавание гранулометрического состава шарообразных тел применительно к горно-рудному производству. № 5.

Гриняк В. М., Можаровский И. С., Дегтярев К. И. Нейросетевая модель планирования сезонных продаж. № 7.

Исхаков А. Р., Асадуллин Р. М., Богданов М. Р., Федоров Н. И. Автоматизация предварительной обработки картографического материала, содержащего цветные контуры в целях их дальнейшей векторизации. № 5.

Каширина И. Л., Львович Я. Е., Тузиков А. А. Нейросетевое резервирование дублированных измерений параметров при наземных огневых испытаниях жидкостных ракетных двигателей. № 9.

Красовская Л. В. Имитация отработки действий на нечетких семантических сетях в алгоритмах самообучения интеллектуальных систем. № 11.

Куравский Л. С., Мармалюк П. А., Панфилова А. С., Ушаков Д. В. Исследование факторных влияний на развитие психологических характеристик с применением нового подхода к оценке адекватности моделей наблюдениям. № 11.

Новикова Н. М., Ляликова В. Г. Математические модели параметрических статистических и нейросетевых обнаружителей сигналов при наличии шума и импульсной помехи. № 5.

Осипов В. Ю. Устойчивость рекуррентных нейронных сетей с управляемыми синапсами. № 9.

Потапов В. И. Модели и задачи оценки надежности нейронной системы при обучении и переобучении нейронной сети после отказов. № 11.

Скрибцов П. В., Казанцев П. А., Долгополов А. В. Виртуальные нейросетевые процессоры — метод кроссплатформенного ускорения расчета нейросетевых задач на современных многоядерных процессорах. № 1.

Скрибцов П. В., Казанцев П. А., Долгополов А. В. Особенности реализации алгоритмов распознавания объектов на фото и видео с применением современных многоядерных процессоров. № 3.

Четырбоцкий А. Н. Параметрическая идентификация радиальных базисных функций нейронных сетей методами глобальной оптимизации. № 11.

Указатель приложений к журналу "Информационные технологии", опубликованных в 2011 г.

Актуальные проблемы в исследованиях Института информационных технологий НАН Азербайджана. № 9.

Буянкин В. М., Гоменюк С. М., Карпенко А. П., Литун Т. О. Нейросетевая идентификация и управление некоторыми механизмами параллельной кинематики. № 11.

Королёв В. И., Новиков А. А., Шарков А. Е. Электронное правительство: анализ, концептуальное представление, функциональные приложения. № 8.

Кудинов И. Ю. Интеллектуальные технологии моделирования и управления многосвязными объектами. № 3.

Кухаренко Б. Г. Алгоритмы анализа изображений для определения локальных особенностей и распознавания объектов и панорам. № 7.

Кухаренко Б. Г. Сегментирование изображений на основе сечения графов, марковских случайных полей и алгоритмов разведывания данных. № 2.

Майданович О. В., Охтилев М. Ю., Соколов Б. В., Юсупов Р. М. Комплексная автоматизация мониторинга состояния космических средств на основе интеллектуальных информационных технологий. № 10.

Малыхин В. М., Малыхина Г. Ф., Меркушева А. В. Концепция, методы, алгоритмы и нейросетевые структуры для восстановления формы сигналов при регистрации их смеси. № 1.

Памяти А. С. Нариньяни. № 6.

Сигарев А. А. Концепция построения многопроцессорных вычислительных систем с массовым параллелизмом с линейным ростом ускорений. № 12.

Трахтенгерц Э. А. Информационные технологии формирования управленческих решений в процессе манипулирования общественным мнением. № 5.

Штрик А. А. Критерии, показатели и методики оценки эффективности электронных правительств. № 4.

CONTENTS

Vasenin V. A., Shundeev A. S. *Development of Web Technology and Middleware* 2

This article provides an overview of these technologies, analyze their impact on the construction of middleware software, as well as forecasts of their future development.

Keywords: middleware, web-service, REST

Tarnavsky G. A., Chesnokov S. S. *Client Base Analysis of Computer Simulation Center, the Cloud Portal of Internet* 9

The function experience of Computer Simulation Center in Internet is analysed. Visit statistics of Web-portal, the pioneer project based on advanced information technologies of cloud computer trend and intended to commercial finalization of conducted program treatments, are considered.

Keywords: Internet, informational technologies, computer simulation, cloud computing, statistics of visits

Rogozov Yu. I., Sviridov A. S., Degtyarev A. A. *Analysis and Prospects of Software Development* 16

The paper conducted a historical analysis of approaches to software development, outlines key races, gives a brief description of each approach. The urgency of Research and Development Approach Model Based Software Engineering

Keywords: software development, programming languages OOP CBSE MDE, MBSE

Lidovski V. V. *Analysis of LR-Parsing for 18 Programming Languages* 21

The actuality of the usage of canonical LR(1)-tables are explored. The results of the tests of the grammars of 18 widely used programming languages are provided. The conclusion about usefulness of support of LR(1)-parsing by compiler compilers is given.

Keywords: compiler, parser, LR, LALR, Bison, Yacc

Derbenev N. V., Tolcheev V. O. *Using a Method of Detecting Near Duplicates in Sciencemetric Analysis* 24

We develop and research a method of detecting near duplicates in the articles of scientific journals. We apply the method for sciencemetric analysis. We consider a way of forming sampling for research and define a threshold, which divides original publications and near duplicates.

Keywords: near duplicates, sciencemetric analysis, Jaro-Winkler similarity, shingling, coefficients of similarity, Jaccard coefficients

Yusupova N. I., Valeev R. S. *Rational Placement of Cargoes into Containers Considering their Physical Characteristics with Use of the Robotized Complex* 30

The problem of placement of various cargoes into containers which occurs at the performance of different logistics operations which are made with use of the industrial robot-loader is considered in this paper. The mathematical two-criteria problem statement in account of the placement density and requirements to the center of gravity location is presented. For solving this problem the algorithm of dense placement of cargoes into containers, the evolutionary algorithm of the search of the feasible center of gravity location of loaded container considering physical characteristics of cargoes, the method for finding the center of gravity of cargoes with use of robotized complex are developed.

Keywords: dense placement of cargoes, robot-loader, center of gravity, force- torque sensor, evolutionary algorithm, multilevel method

Bolkhovityanov A. V., Gusev S. V., Chepovskiy A. M. *Model and the Knowledge Base of Verbal Syntactic Relation in Russian Sentence* 37

This paper proposes the model that is used to describe relations between verb and set expressions during analysis of the sentence in the Russian language. It describes the developed knowledge base that is used to describe syntactic units of the Russian language. The model is used to objects identification and knowledge extraction from the text in Russian language.

Keywords: natural language processing, syntactic relation between verb and dependent words, syntactic analysis

Kuznetsov L. A. *Status Assess Measures of Objects and Processes for Monitoring* 40

Developed status assesses measures of objects and processes, monitoring of which are ensured by the supervision of a set of random variables. Within monitoring measured values of the set of physical quantities of different nature. The developed methodology provides a structure of the controlled variables as a vector

of the monitoring, realization of which formed the characteristics assessment and evolution of objects and processes. For the states and the evolution of the object represented by a set of random controlled variables, we introduce the scalar characteristics which values are determined by retrospective data monitoring. Characteristics allow to evaluate the actual state of the object and its trends, and can be used in the synthesis of automated monitoring systems, objects and processes.

Keywords: condition monitoring, random vector, covariance matrix, the entropy

Ivanov I. A., Uvaysov S. U. Information Model of Process of Designing Testability Radio-Electronic Equipment 45

In this work the information model of testability procurement process of radio-electronic equipment at the design phase is presented and this is a complex of interconnected moduluses for importation, processing, storage and transfer of data. This model contains all basic properties and data communication of object by modern automated design with a glance of heterogeneity physical process.

Keywords: designing radio-electronic equipment, information model, testability, diagnostication, modeling electric, thermal, mechanical, diagnostic

Speransky D. V. State Identification of Some Type of Bilinear Systems 48

Object of study is a special type of bilinear systems. Criteria for the existence of synchronizing, homing and diagnostic sequences of the above mention systems are obtained. These criteria in terms of characteristic matrices are formulated. Analytics methods for finding the above mention sequences are obtained and prove.

Keywords: bilinear systems, synchronizing sequence, homing sequence, diagnostic sequence, the criteria of existence, the analytical methods of construction

Frolov S. V., Gorbunov A. V., Lyadov M. A., Farea S. G. Block-Hierarchical Approach Used for Creating a Knowledge Base of an Information System to Monitor the State of Human Brain 54

It is suggested to use knowledge bases built on the base of the Boolean algebra in information monitoring system to predict the critical states of complex objects. The problem of high dimensionality of truth tables is solved by using a block-hierarchical approach. Knowledge base information system was created to monitor the state of the human brain. This system is able to predict the possibilities of cerebral stroke when it can be prevented.

Keywords: Boolean algebra, block-hierarchical approach, knowledge base, information system of monitoring, complex system, critical state of a complex object, state of the human brain, cerebral stroke.

Sadykov S. S., Belyakova A. S. Mathematical Models of Some Cardiovascular Diseases 59

Mathematical models of three cardiovascular diseases are constructed. Their work at change of parameters of functioning of heart is investigated.

Keywords: cardiovascular disease, parameter, correlation, a mark, regression analysis, importance, model, informative

Kovács G., Shilov V. V. M-3: On the History of the First Generation Computer 73

Memoirs of Dr. Győző Kovács, who was a member of team, which constructed the first Hungarian electronic computer M-3 following the basic design of Soviet M-3 computer with the comments and additional materials on the history of M-3 computer in other countries.

Keywords: Cybernetics Research Group of the Hungarian Academy of Sciences, M-3 computer, first computer in Hungary, first computer program in Hungary, magnetic drum, first computer centre in Hungary

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Дизайнер *Т.Н. Погорелова*. Технический редактор *Е.В. Конова*.

Корректор *М.Г. Джавадян*.

Сдано в набор 05.10.2011. Подписано в печать 17.11.2011. Формат 60×88 1/8. Бумага офсетная. Печать офсетная.

Усл. печ. л. 9,8. Уч.-изд. л. 11,58. Заказ 827. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Отпечатано в ООО "Подольская Периодика"

142110, Московская обл., г. Подольск, ул. Кирова, 15