

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Пензенский государственный университет» (ПГУ)
Российский фонд фундаментальных исследований

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СИСТЕМЫ

СБОРНИК НАУЧНЫХ СТАТЕЙ
XV Международной научно-технической
конференции,
посвященной 75-летию Пензенского государственного университета

г. Пенза, 20–21 ноября 2018 г.

Proceedings of the Fourteenth International Conference
of Science and Technology

NEW INFORMATION TECHNOLOGIES
AND SYSTEMS
dedicated to the 75th anniversary of Penza state university

Penza, Russia, November 20–21, 2018

Пенза

УДК 004.4
Н76

Новые информационные технологии и системы : сб. науч. ст.
Н76 XV Междунар. науч.-техн. конф., посвящ. 75-летию Пенз. гос. ун-та
(г. Пенза, 20–21 ноября 2018 г.). – Пенза : Изд-во ПГУ, 2018. – 242 с.

ISBN 978-5-907102-42-2

Изложены проблемы в области аппаратно-программного обеспечения информационно-вычислительных систем и систем управления, систем автоматизированного проектирования. Рассматриваются вопросы управления в социальных, экономических и технических системах. Представлены современные технологии хранения и обработки данных, информационные технологии в образовании. Обсуждаются вопросы информационной безопасности вычислительных систем и сетей, интеллектуальных систем и машинного обучения, сетей ЭВМ и телекоммуникаций.

Издание предназначено для инженеров и научных работников, а также обучающихся старших курсов, интересующихся вопросами современных информационных технологий.

УДК 004.4

Редакционная коллегия:

В. И. Волчихин, Д. В., И. И. Артемов, Г.В.Козлов, Л.Р. Фионова, Г.В. Бобрышева, В.И. Горбаченко, А.Ю. Старикова

Проведение конференции поддержано по результатам конкурсного отбора научных проектов в конкурсе Г - Конкурс проектов 2018 года организации российских и международных научных мероприятий, проводимый РФФИ (научный проект № 18-07-20092)

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СИСТЕМ УПРАВЛЕНИЯ

А. В. Чернышов

Мытищинский филиал МГТУ им. Н. Э. Баумана, Мытищи, Россия

ПРОБЛЕМА ПРОЕКТИРОВАНИЯ АРХИВНЫХ ХРАНИЛИЩ ЭЛЕКТРОННОЙ ИНФОРМАЦИИ

Рассмотрена проблема поиска оптимальной структуры долговременного электронного архивного хранилища информации, включающая как минимизацию вероятности потери хранимой информации, так и минимизацию стоимости создания архива.

С развитием информационных технологий всё большее количество документов создаётся и используется исключительно в электронном виде. Как и в случае с бумажными документами, вышедшие из активного использования документы должны быть направлены в архив для длительного хранения с возможностью доступа к ним при необходимости. На сегодняшний день интерес к архивам длительного хранения электронной информации проявляют также такие организации как музеи и фундаментальные библиотеки, ведущие оцифровку своих фондов.

Согласно действующему ГОСТ [1], долговременное архивное хранение электронных документов должно осуществляться исключительно на носителях типа WORM, к которым на сегодняшний день относятся только оптические диски однократной записи (CD-R, DVD±R, BD-R и подобные).

Для удобного управления большими массивами оптических дисков, включая оперативный доступ к записанной на них информации, создаются и эксплуатируются роботизированные библиотеки оптических дисков различной конструкции [2, 3, 4], активно используемые для создания архивных хранилищ электронной информации [5, 6].

Однако до настоящего момента вопрос определения надёжности хранения информации в проектируемом или существующем хранилище остаётся неразработанной научной проблемой.

Как известно, если задаться вероятностью выхода из строя одного оптического диска q , что будет соответствовать потери информации на этом диске, то вероятность потерять хотя бы часть информации в массиве из l дисков, зарезервированных n копиями, составит:

Вторым важным параметром при проектировании архивного хранилища является суммарная стоимость хранения заданного объёма информации W , определяемая, с одной стороны, ёмкостью применяемых оптических дисков (i , как следствие, их количеством l), а с другой стороны, стоимостью c одного оптического диска данного типа:

Здесь дополнительно учтена стоимость используемого привода оптического диска Sp . Данная формула применима при оценке первоначальных затрат на создание архива. В ней не учтена стоимость оптических дисков, которые должны будут использоваться для замены дисков, вышедших из строя в процессе хранения, количество которых в конечном итоге будет определяться значением q .

Сразу оговоримся, что здесь и в дальнейшем будет рассматриваться только проблема создания локального архивного хранилища. Причины, по которым нежелательно использовать для этих целей облачные технологии, подробно рассмотрены в [7].

Как можно видеть, для определения параметров проектируемого архивного хранилища знание значения параметра q жизненно необходимо. Однако в настоящий момент проектировщики архивов вынуждены отталкиваться от совсем других показателей, обычно предоставляемых производителями оптических дисков и создателями роботизированных библиотек, заявляющих о 30-50-100 годах «срока службы» оптических дисков.

Проблема связана, во-первых, с тем, что заявляемый срок службы определяется, как правило, по ускоренным тестам [8], т.е. является больше теоретически вычисленным, чем полученным в результате практических испытаний, а во-вторых, с тем, что ускоренные тесты проводятся на выборочных партиях дисков и их, вообще говоря, нельзя однозначно экстраполировать на все партии этих дисков. Кроме этого, сама сущность параметра «срок службы» не позволяет определить по нему значение вероятности выхода из строя q .

В принципе понятно, что при проектировании архива необходимо добиться хотя бы соотношения $Q_n < q$, а для архивов наиболее ценной информации $Q_n < 0,1q$ или даже $Q_n < 0,01q$. Понятно, что этого можно достигнуть, увеличивая параметр n . В работе [9] показано, что минимизация стоимости архива может быть достигнута выбором типа оптических дисков. Однако итоговый результат будет дополнительно сильно зависеть от предполагаемого к хранению в архиве объёма информации W .

Дальнейшее уменьшение стоимости архива может быть достигнуто отходом от простого создания n запасных копий каждого оптического диска и организацией дисков в отказоустойчивые структуры типа RAID5, RAID6 [10], что позволяет значительно уменьшить количество запасных копий n получившихся структур для сопоставимого итогового значения Q . Здесь, однако, возникает сразу несколько проблем.

1. Проблема обеспечения оперативности доступа к хранимой информации.
2. Проблема определения оптимального количества дисков в одной структуре RAID.
3. Проблема выбора оптимальной структуры RAID.

Эти проблемы взаимосвязаны и во многом зависимы от общего объёма информации, хранящегося в архиве. Так для относительно небольшого архива (например, музей регионального или местного значения с предполагаемым объёмом архива 5–10 Тбайт) дилема между минимизацией стоимости архива, надёжностью хранения информации, скоростью и удобством доступа будет представлять серьёзную проблему оптимизации, поскольку применение готовых промышленных решений (типа роботизированной библиотеки Panasonic LB-DH8 [2]) невозможно из-за высокой стоимости как приобретения (порядка 6 млн руб.), так и владения (около 10 % от первоначальной стоимости в год) [6]. Отказ же от применения роботизированной библиотеки предполагает организацию возможности записи и чтения оптических дисков в структуре RAID на отдельном компьютере, что предполагает одно из следующих решений:

- установка на компьютере необходимого количества приводов оптических дисков, что обеспечивает относительно высокую скорость и удобство доступа к информации (просто необходимо установить диски соответствующего массива в приводы), но увеличивает первоначальную стоимость архива на суммарную стоимость приводов;
- установка на компьютере одного привода оптических дисков, но использование жёсткого диска компьютера в качестве промежуточного носителя для последовательного считывания образов оптических дисков, образующих массив RAID, и последующей сборки массива на компьютере из считанных образов, что минимизирует затраты на приобретение приводов оптических дисков (жёсткий диск необходимой ёмкости скорее всего уже входит в состав компьютера), но сильно увеличивает время доступа к информации (до чтения первого байта из массива необходимо считать на жёсткий диск компьютера с оптических дисков весь объём массива RAID) и значительно снижает удобство доступа;
- отказ от использования структур RAID, использование одиночных дисков, что гарантирует возможность прямого считывания любой информации всегда с одного оптического диска (нет необходимости иметь дополнительные приводы, нет необходимости в предварительном считывании содержимого всего диска), но для достижения необходимого значения Q требует хранения в архиве большого количества запасных копий n ;

- использование комбинации одиночных дисков и структур RAID, что, теоретически, должно обеспечить прямой доступ к информации (на одиночных дисках) и одновременно минимизировать количество копий n (за счёт избыточности RAID).

Для архивов большого объёма (100 Тбайт и больше, как правило, не имеющих проблем с приобретением дорогостоящего оборудования) выбор в качестве основы для построения современных роботизированных библиотек также не является однозначным. Чем больше объём хранимой информации, тем меньше вероятность того, что какая-то конкретная часть этой информации будет запрошена. Известно, что даже для информации, хранящейся на дисках первого уровня (т.е. в оперативном доступе), около 80 % данных не требуют частого обращения, а порядка 20 % из них никогда не будут запрошены [11]. Соответственно, с точки зрения экономии затрат на содержание архива, может оказаться гораздо выгоднее хранить диски с редко запрашиваемой информацией не в роботизированных библиотеках, а на обычных стеллажах, имея специальную роботизированную библиотеку, в которую диски с запрошенной информацией устанавливаются при необходимости вручную.

Количество дисков, включаемых в структуру RAID, теоретически, неограничено. И поскольку в структурах RAID специально предусмотрены «избыточные» диски (1 для RAID5, 2 для RAID6), увеличивающие стоимость единицы хранения информации, понятно, что с увеличением общего количества дисков в массиве, стоимость единицы хранения информации снижается. Однако при этом будет расти вероятность потери информации, что может быть компенсировано увеличением количества запасных копий n , что приведёт к увеличению стоимости архива.

Не менее сложной проблемой является собственно выбор между структурами RAID5 и RAID6. Дополнительно могут быть рассмотрены комбинированные структуры типа RAID15, RAID16. Какие-либо другие структуры, не являющиеся классическими, для организации долговременных архивов рассматривать не имеет смысла, так как невозможно гарантировать поддержку этих неклассических структур в течение длительного времени, что автоматически породит проблемы с воспроизведением оптических дисков, записанных в этих структурах, с течением времени.

Таким образом, проблема проектирования архивных хранилищ электронной информации может быть формализована следующим образом.

Исходные данные (задано):

- максимальный объём информации в архиве W ;
- максимально допустимая вероятность потери хотя бы части информации Q ($Q < q_i$ или $Q < 0,1q_i$, или $Q < 0,01q_i$);
- доступны для использования оптические диски однократной записи типов 1..k;
- вероятность выхода из строя одного оптического диска q_i ($i=1..k$);
- стоимость одного оптического диска c_i ($i=1..k$);
- допускается применение структур: одиночные диски, RAID5, RAID15, RAID6, RAID16 или их комбинаций (обозначим этот параметр как S);
- стоимость оборудования (например, привода или роботизированной библиотеки) для записи/считывания оптических дисков C_p (величина этого параметра сильно зависит от конкретного типа применяемого оборудования и используемых структур данных, здесь не конкретизируется). На выбор значения этого параметра сильное влияние оказывает такой плохо формализуемый фактор, как требование степени удобства и скорости доступа к информации (в зависимости от типа архива, а также имеющихся финансовых возможностей, заказчик может пойти на значительное снижение скорости и удобства доступа при условии значительного удешевления итогового архива).

Требуется:

$$Q(W, i, n, S) \rightarrow \min;$$

$$C(W, i, n, S, C_p) \rightarrow \min.$$

Здесь параметр i определяет использование соответствующих значений q_i и c_i .

Совместная минимизация Q и C и даст решение проблемы проектирования архивного хранилища электронной информации с оптимальными параметрами, соответствующего требованиям современного ГОСТ [1]. Однако сделать это будет возможно только после определения достоверного значения параметра q, что требует отдельного исследования и выходит за рамки данной статьи.

Библиографический список

1. ГОСТ Р 54989–2012 / ISO TR 18492:2005. Обеспечение долговременной сохранности электронных документов (вступил в силу: 01.05.2013).
2. Data Archiver LB-DH8 series. – URL: <http://panasonic.net/avc/archiver/lb-dh8/> (дата обращения: 01.02.2016).
3. New Sony Everspan Library System Delivers Reliable Optical Disc Archiving For Data Centers // CDRinfo. – URL: <http://www.cdrinfo.com/Sections/News/Details.aspx?NewsId=45763> (дата обращения: 14.05.2016).
4. Optical Jukeboxes and Libraries // KINTRONIX: IP Security Solutions. – URL: <https://kintronics.com/solutions/optical-jukeboxes-and-libraries/> (дата обращения: 14.05.2016).
5. Робот-библиотекарь поселился в новом кампусе СПбГУ «Михайловская дача» // Санкт-Петербургский государственный университет. – URL: <http://spbu.ru/smi/o-nas-pishut/24726-robot-bibliotekar-poselilsya-v-novom-kampuse-spbgu-mikhajlovskaya-dacha.html> (дата обращения: 13.11.2015).
6. Российские организации начинают строить электронные архивы на оптических дисках // Ассоциация электронных торговых площадок. – URL: <http://www.aetp.ru/market-news/item/400867> (дата обращения: 13.11.2015).
7. Каташин, В. В. Проблема надёжного хранения файлов в архивных электронных хранилищах небольших организаций (на примере музеев) / В. В. Каташин, А. В. Чернышов // Перспективы развития информационных технологий : сб. материалов XIX Междунар. науч.-практ. конф. – Новосибирск : ЦРНС, 2014. – С. 12–23.
8. ISO/IEC 10995:2011(E) Information technology – Digitally recorded media for information interchange and storage – Test method for the estimation of the archival lifetime of optical media. – 28 p.
9. Чернышов, А. В. К вопросу о применении оптических дисков для создания долговременных электронных архивных хранилищ информации небольших организаций / А. В. Чернышов // Информационные технологии. – 2016. – Т. 22, № 8. – С. 635–640.
10. Чернышов, А. В. Модель надёжности хранения информации на современных библиотеках оптических дисков, объединенных в массивы RAID-6 / А. В. Чернышов // Вестник МГТУ им. Н. Э. Баумана. Сер.: Приборостроение. – 2017. – № 3. – С. 65–75.
11. Корепанов, И. Как сохранить архив на десятилетия? / И. Корепанов // Журнал сетевых решений/LAN. – 2008. – № 03. – URL: <http://www.osp.ru/lan/2008/03/4899898> (дата обращения: 04.11.2015).

В. В. Бурлин, Л. В. Гурьянов

Пензенский государственный университет, Пенза, Россия

О РЕАЛИЗАЦИИ АЛГОРИТМА БЕРНШТЕЙНА–ВАЗИРАНИ НА ЯЗЫКЕ PYTHON

Рассматривается программное средство, реализующее поиск числа a в функции $f(x)_a = ax \bmod 2$ с помощью алгоритма Бернштейна–Вазирани.

Задача Бернштейна–Вазирани заключается в следующем: имеется n -битная функция $f: \{0,1\}^n \rightarrow \{0,1\}$, которая возвращает один бит, эта функция гарантированно имеет вид: $f_s(x) = xs$, где s – это неизвестная n -битная строка и $xs = x_1s_1 + x_2s_2 + \dots + x_ns_n \bmod 2$. Задача Бернштейна–Вазирани – это поиск неизвестной строки s .

Классический оракул возвращает $f_s(x) = xs \bmod 2$, с учётом ввода x . Квантовый оракул ведёт себя аналогично, но может быть опрошен с суперпозицией x . Классически, скры-

тое число может быть обнаружено путём опроса оракула с $x=1, 2, \dots, 2^i, \dots, 2^{n-1}$, где каждый запрос показывает i -й бит числа s . Например, с помощью $x=1$, может быть получен наименее значимый бит числа s и так далее. Это оказывается оптимальной стратегией; любой классический алгоритм, который с большой вероятностью находит скрытое целое число, должен запросить оракул $\Omega(n)$ раз. Однако, учитывая соответствующий квантовый оракул, скрытое целое число можно найти только с помощью одного запроса, используя алгоритм Бернштейна-Вазирани.

Квантовый алгоритм для задачи Бернштейна-Вазирани, выявления скрытого целого, следует естественным путем запроса квантового оракула f_a с квантовой суперпозицией, полученной из преобразования Адамара над $|0\rangle$, а именно:

$$|0\rangle \xrightarrow{\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle} \xrightarrow{f_a} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{a^*x} |x\rangle.$$

Квантовый оракул f_a возвращает 1 на входную строку x , когда $a^*x \equiv 1 \pmod 2$, и возвращает 0 в противном случае. Для создания алгоритма требуется создать оракула. Можно заметить, что

$$(-1)^{ax} = (-1)^{a_1x_1} \dots (-1)^{a_i x_i} \dots (-1)^{a_n x_n} = \prod_{i:a_i=1} (-1)^{x_i}.$$

Следовательно, оракул внутреннего произведения может быть реализован следующим унитарным преобразованием, которое разложимо как одноквантовые унитарные:

$$U_{f_a} = U^1 \otimes U^2 \otimes \dots \otimes U^i \otimes \dots \otimes U^n,$$

где

$$U^i = \left\{ \begin{array}{l} I, \quad a_i = 0 \\ Z, \quad a_i = 1 \end{array} \right\}, I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad [1]$$

или

$$U^i = (1 - a_i)I + a_i Z,$$

где Z – это Z -матрица Паули, I – единичная матрица для $a_i \in \{0,1\}$.

Общая квантовая схема показана на рис. 1.

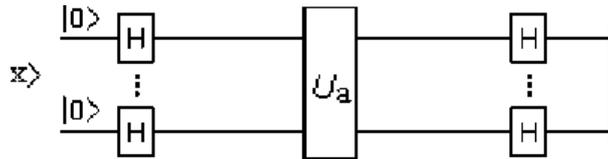


Рис. 1. Квантовая схема для алгоритма Бернштейна-Вазирани

Программное обеспечение реализации алгоритма Бернштейна-Вазирани представляет собой набор файлов исходного кода, написанных на языках программирования Python [2], и разработано на основе фреймворка QISKit. QISKit – это набор программных средств для работы с квантовыми схемами короткой глубины и построения приложений ближнего порядка и экспериментов на квантовых компьютерах. В QISKit квантовая программа представляет собой массив квантовых схем.

Каталог qiskit является основным модулем Python и содержит объекты интерфейса программирования QuantumProgram, QuantumRegister, ClassicalRegister и QuantumCircuit.

На самом высоком уровне пользователи определяют QuantumProgram для создания, изменения, компиляции и выполнения набора квантовых схем. Каждый QuantumCircuit

имеет набор регистров данных, каждый из которых является QuantumRegister или ClassicalRegister. Методы этих объектов используются для исполнения инструкций, определяющих схему. Затем QuantumCircuit генерирует код OpenQASM, который может передаваться другим компонентам в каталоге qiskit.

Квантовые схемы взаимодействуют с помощью программных компонент следующим образом (рис. 2).

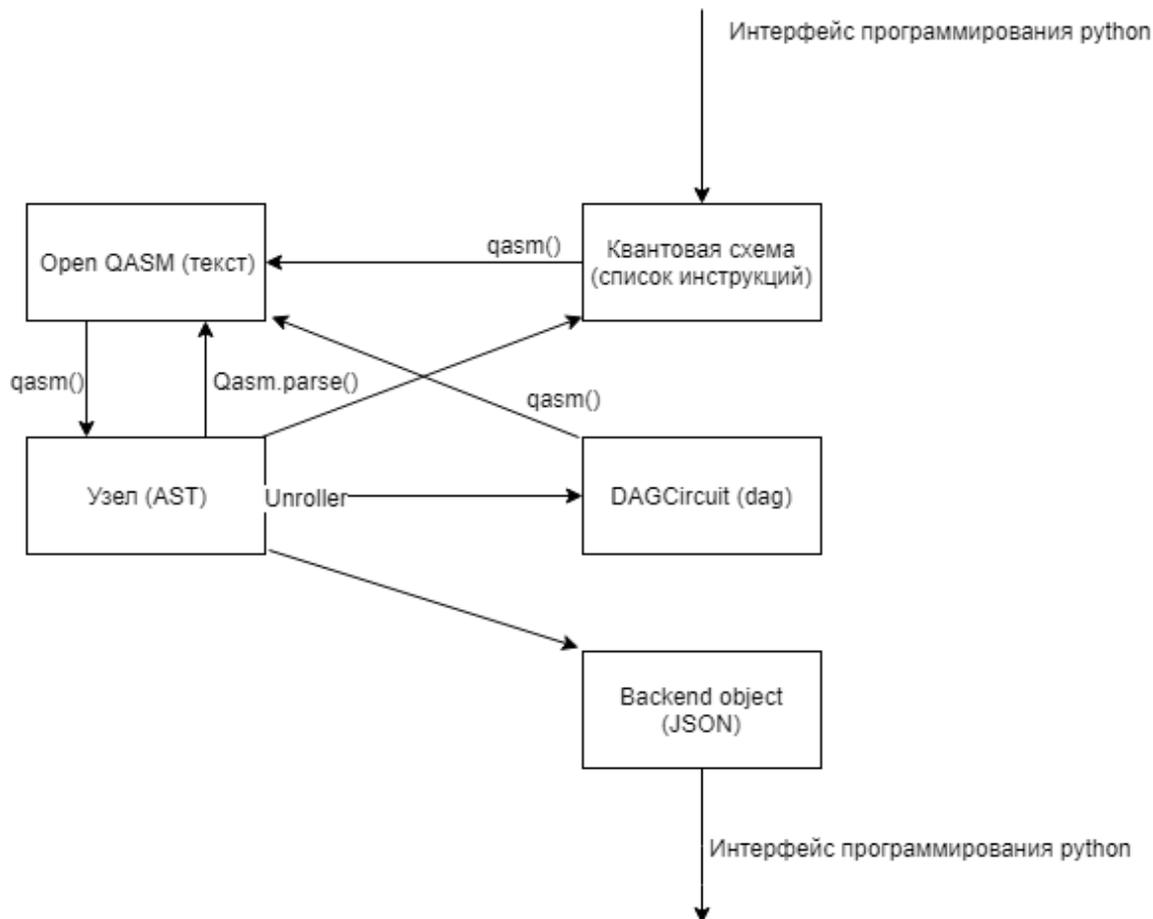


Рис. 2. Квантовые представления и как они трансформируются друг в друга

Интерфейс программирования используется для создания схем OpenQASM для текстовых или QuantumCircuit объектов. Источник OpenQASM, как файл или строка, передается в объект Qasm, метод parse которого создает абстрактное синтаксическое дерево (AST). AST передается в Unroller, который подключен к UnrollerBackend. Существуют версии PrinterBackend для вывода текста, JsonBackend для создания ввода в симулятор и экспериментальные серверы, DAGBackend для построения объектов DAGCircuit и Circuit Backend для создания объектов QuantumCircuit. Объект DAGCircuit представляет собой «развернутую» схему OpenQASM как ориентированный ациклический граф (DAG). DAGCircuit предоставляет методы для представления, преобразования и вычисления свойств схемы и вывода результатов, как OpenQASM. Весь поток используется модулем target для перезаписи схемы для выполнения на устройстве с фиксированными соединениями, определенными в CouplingGraph.

Функциональное тестирование показало корректность работы разработанных программных средств.

Библиографический список

1. CSE 599d – Quantum Computing. Winter 2006. Instructor: Dave Bacon. – URL: <http://courses.cs.washington.edu>
2. УэсМаккинни. Python и анализ данных / УэсМаккинни. – М. : ДМК Пресс, 2015. – 482 с.

Р. Н. Федюнин

Пензенский государственный университет, Пенза, Россия

АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ ДЛЯ ОБРАБОТКИ ОДНОРОДНЫХ МАТРИЦ

Систолическая архитектура вычислительного устройства является решением, агрессивно эксплуатирующим параллелизм «конвейерного» типа. Несмотря на дороговизну решения, систолы являются одним из самых эффективных методов решения специализированных задач, так как часто обеспечивают на конкретной задаче быстродействие, не достижимое для большинства микропроцессорных архитектур.

Систолические системы являются специализированными вычислителями и производятся в основном под конкретную задачу. Фактически, задача построения систолического вычислителя сводится к построению аппаратного конвейера, имеющего достаточно большое время получения результата (т.е. большое количество ступеней) но при этом сравнительно маленькое время между последовательной выдачей результатов, так как значительное количество промежуточных значений обрабатывается на разных ступенях конвейера.

К достоинствам систолической структуры относятся следующие: систолы отличаются простой и регулярной структурой, вычисления в систолах «ритмичны», последовательны, экономически выгодно построение больших партий, вычислительная эффективность систол определяется правильным построением конвейера.

Анализ подходов организации вычислений, способов управления вычислениями в систолических арифметико-логических процессорах [1] показывает, что для выполнения массивных операций этого типа наиболее эффективна топология информационных межпроцессорных каналов типа «решетка». Это, прежде всего, связано с алгоритмическими особенностями традиционных методов вычислений и организацией структур данных. Такая топология межпроцессорных связей обеспечивает возможность реализации арифметико-логических операций на базе простого одноразрядного процессорного модуля. Достоинством технических решений является возможность использования для построения операций над данными классических методов, что в свою очередь позволяет интегрировать процессоры в состав классических систем обработки данных без существенного изменения их математической базы.

В качестве процессорного модуля систолического вычислительного пространства для выполнения арифметико-логических вычислений над данными, представляемыми n -разрядными двоичными векторами, достаточно эффективен процессорный элемент (ПЭ) [2, 3] организация которого представлена формулами 1 и 2.

Функциональные возможности процессорного модуля с данной организацией обеспечивают реализацию системы логических функций от пяти переменных. Аргументами функций являются значения логических переменных, поступающих на обработку по информационным каналам t , x , s , p , y , вычисленные значения функций формируются соответственно на информационных выходах процессорного модуля t_{out} , x_{out} , s_{out} , p_{out} , y_{out} . Настройка оборудования модуля ПЭ на выполнение соответствующего типа функций выполняется по каналам управления k , z_1 , z_2 .

Для настройки модуля ПЭ на реализацию требуемой операции достаточно сформировать на его управляющих входах k , z_1 , z_2 значение кода соответствующей операции.

При формировании на управляющем входе процессорного модуля сигнала $k = 1$ обеспечивается реализация следующей системы функций алгебры логики

$$\begin{aligned} s_out &= s \cdot p; \\ p_out &= s \vee p; \\ x_out &= z_1 \cdot (z_2 \cdot x \vee z_2 \cdot t); \\ t_out &= z_1 \cdot z_2 \cdot x \cdot t \vee z_1 \cdot z_2 \cdot (x \vee t) \vee z_1 \cdot z_2 \cdot t; \\ y_out &= y \end{aligned} \quad (1)$$

При формировании на управляющем входе процессорного модуля сигнала $k = 0$ обеспечивается реализация следующей системы функций алгебры логики:

$$\begin{aligned} s_out &= s \cdot y \vee A \cdot y; \\ p_out &= B \cdot y \vee p \cdot y; \\ x_out &= t; \\ t_out &= x; \\ y_out &= y \\ A &= \bar{x} \cdot \bar{s} \cdot \bar{p} \vee \bar{x} \cdot \bar{s} \cdot p \vee \bar{x} \cdot s \cdot \bar{p} \vee x \cdot s \cdot p; \\ B &= \bar{x} \cdot s \cdot p \vee x \cdot \bar{s} \cdot p \vee x \cdot s \cdot \bar{p} \vee x \cdot s \cdot p \end{aligned} \quad (2)$$

Данная система функций описывает арифметические операции, на примере умножения двух целых чисел.

Арифметическое умножение двоичных чисел выполняется в вычислительной среде следующим образом. Множимое поразрядно подается на входы x_1, x_2, \dots, x_n процессорных модулей первого столбца, а разряды множителя устанавливаются на входы y_1, y_2, \dots, y_n процессорных модулей первой строки вычислительной среды. Причем, на входе x первого модуля ПЭ первого столбца устанавливается младший разряд множимого, а на входе y – младший разряд множителя. На остальные входы вычислительной среды устанавливаются значения сигналов, равные 0.

Таким образом, на выходах s_out процессорных модулей столбцов, на входы у которых поступает разряд множителя равный 0, формируется копия числа, установленная на входах s , а на выходах x_out – сдвинутая на один разряд копия числа, установленного на входах x . При установке на входах y процессорных модулей столбца единичного значения разряда множителя на выходах s_out формируется сумма значений данных, установленных на входах s и x модулей ПЭ, а на выходах x_out – сдвинутая на один разряд вниз копия числа, установленного на входах x .

Пример выполнения операции умножения двух положительных двоичных чисел в вычислительной среде из данных процессорных модулей показан на рис. 1.

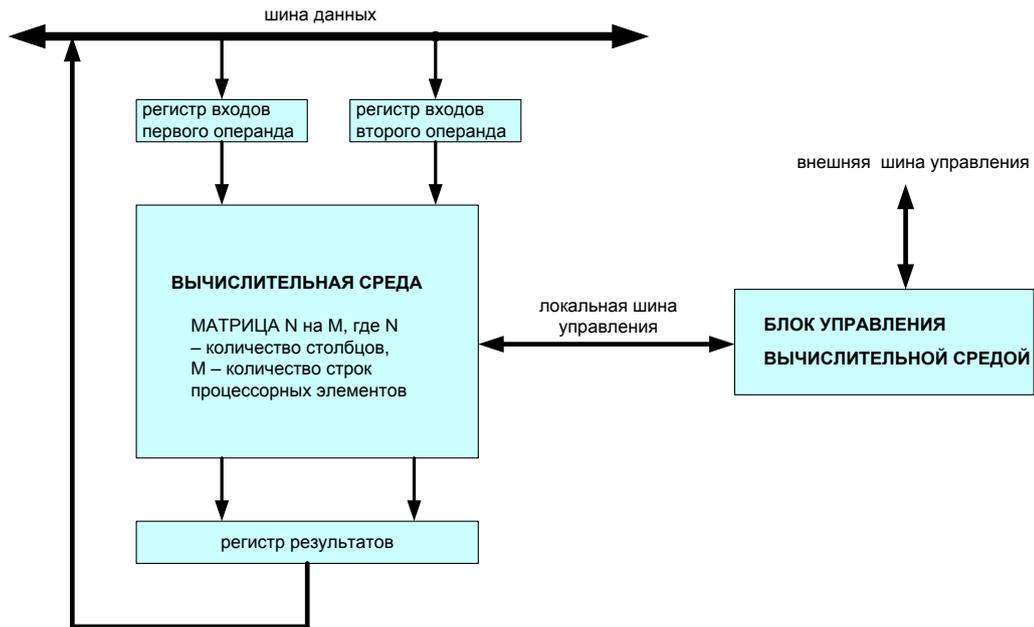


Рис. 1. Процессор на базе систолической структуры

В данном случае множимое подается на группу входов X , а множитель – на группу входов Y . Результат операции умножения формируется на группе выходов S_OUT , начиная с модуля ПЭ первой строки. На соответствующих информационных выходах s_out ячеек первого столбца формируется первое частичное произведения, а на x_out – сдвинутая на один разряд копия множимого. На информационных выходах s_out ячеек второго столбца формируется второе частичное произведения, а на x_out – сдвинутая на два разряда копия множимого и т.д.

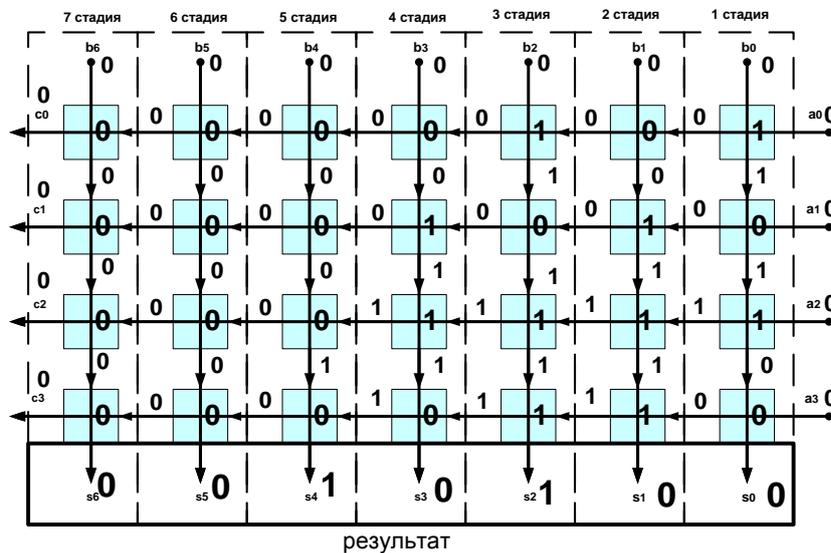


Рис. 2. Процесс реализации умножения в систолической структуре

Выводом данной статьи являются два тезиса. Системы описанного типа являются очень специализированными вычислителями и производятся в основном под конкретную задачу. Фактически, задача построения вычислителя сводится к построению аппаратного конвейера.

Библиографический список

1. Князьков, В. С. Параллельные вычислительные системы с ОКМД-архитектурой : учеб. пособие / В. С. Князьков, Р. А. Бикташев. – Пенза : РИО ППИ, 1991. – 51 с.

2. Пат. 2004136518/09 РФ. Ячейка однородной вычислительной среды / Князьков В. С., Федюнин Р. Н. (от 12 мая 2006 г.).

3. Пат. 2004136518/09 РФ. Ячейка однородной среды / Князьков В. С., Федюнин Р. Н. (от 15 мая 2006 г.).

В. Н. Дубинин, А. С. Войнов, И. В. Сенокосов

Пензенский государственный университет, Пенза, Россия

ПРЕДСТАВЛЕНИЯ 3D-КОНСТРУКЦИЙ В LEGO НА ЯЗЫКЕ ПРОЛОГ И ИХ ПРЕОБРАЗОВАНИЯ

Рассмотрены подходы к представлению 3D-конструкций в LEGO, а также задачи преобразований этих представлений. В качестве языка описания и трансформаций предлагается использовать язык логического программирования Пролог.

Четвертая промышленная революция, более известная как «Индустрия 4.0», первоначально определялась как средство повышения конкурентоспособности обрабатывающей промышленности Германии через усиленную интеграцию киберфизических систем в заводские процессы. В дальнейшем оказалось, что это событие является глобальным и определяет принципиально новый подход к определению свойств всех артефактов, а также методам их производства и потребления [1]. Помимо всего прочего, «Индустрия 4.0» предполагает переход от массового производства (*mass production*) к массовому производству по индивидуальным заказам (*mass customization*). В этом случае изделия определяются самим потребителем. Это в свою очередь требует быстрой разработки техпроцесса изготовления каждого конкретного изделия, а также производственной линии для его производства. Сборка является одной из ключевых областей в производстве. В подавляющем большинстве случаев объекты сборки являются 3D-изделиями, чем и определяется повышенный интерес к 3D-конструированию.

Можно выделить следующие этапы в проектировании производства, ориентированного на массовую кастомизацию:

- 1) определение требований к изделию;
- 2) описание изделия на некотором доменно-ориентированном языке;
- 3) генерация структуры производственной линии и техпроцесса сборки изделия;
- 4) физическая/виртуальная конфигурация производственной линии;
- 5) генерация кода управляющих программ и их загрузка в контроллеры системы.

Во многих случаях этап 4 совершается путем быстрой перенастройки многофункционального производственного оборудования и реконфигурации структуры производственной линии, что возможно на основе использования сетевых технологий и искусственного интеллекта. Зачастую одна и та же производственная линия используется для сборки семейства продуктов, имеющих некоторые общие свойства, называемого также линейкой продуктов. Очевидно, что программное обеспечение быстро реконфигурируемых производственных линий становится сложнее, что требует разработки новых подходов к его разработке.

В данной работе рассматривается этап 2 проектирования. В качестве 3D-изделий рассматриваются конструкции LEGO [2]. Выбор конструктора LEGO определяется его простотой, понятностью и возможностью выявить на LEGO-моделях основные проблемы, возникающие на данном этапе проектирования.

В работе [3] для описания LEGO-конструкций использовались онтологические языки. В данной работе в качестве языка описания предлагается использовать язык логического программирования Пролог ввиду его реляционной природы и наличию механизма логического вывода [4].

Этап 2 предполагает решения следующих задач:

- 1) разработку методов Prolog-описаний различных 3D-фигур и конструкций;
- 2) разработку методов преобразования одной формы описания 3D-фигур в другую форму описания;
- 3) разработку методов сопоставления одной формы описания 3D-фигур с другой форму описания (в частности, для определения, являются ли два различных описания эквивалентными).

Требование к методу описания:

1. Метод описания должен быть как можно более универсальным, т.е. быть пригодным для описания любых 3D-фигур, любого уровня сложности, при любом наборе типов и типоразмеров деталей. В результате применения метода получается описание 3D-фигуры в определенном формате.

2. Получаемые описания на Прологе должны быть полными, то есть полностью описывать LEGO-конструкцию. По этому описанию можно в точности построить ту LEGO-конструкцию, которая описывалась.

3. Получаемые описания на Прологе должны быть корректными и целостными, то есть одна часть описания не должна противоречить другой части того же самого описания. Кроме того, для описания должны выполняться условия корректности и целостности, которые накладываются на LEGO-конструкцию.

4. Получаемые описания на Прологе должны быть компактными, то есть должны быть представлены с помощью как можно меньшего числа фактов и правил, а также числа строк в коде описания.

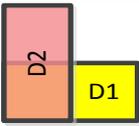
Классификация подходов к описанию приведена ниже.

А. Классификационный признак: Пространственное положение деталей

Подходы:

а) «относительный». В этом случае какая-либо деталь принимается за «точку отсчета» («базу», «базовую деталь») и описание всей 3D-конструкции производится относительно нее, причем понятие координат (x, y, z) вообще не используется, все описание осуществляется только используя положение одной детали одной относительно другой детали (то есть, используя только разнообразные *пространственные отношения*). По мере необходимости можно вводить новые пространственные отношения. Для деталей в виде прямоугольных параллелепипедов со сторонами 2:1 (длина : ширина) выявлено 13 пространственных отношений соединения и 4 пространственных отношения касания.

В таблице ниже приведен пример описания одного из отношений соединения.

| Идент. | Геометрическая интерпретация | Текстовая интерпретация | Формальное обозначение |
|--------|---|--|--------------------------|
| $r1$ |  <p>(вид сверху)</p> | Вторая деталь перпендикулярно лежит на первой, причем первая деталь лежит длинной стороной к пользователю, а вторая деталь находится с левой стороны первой детали и выступает вперед. | $D1r1D2$ или $r1(D1,D2)$ |

б) «абсолютный». В этом случае для определения пространственного положения (иначе, адреса) детали используются «абстрактные» координаты (x, y, z) , причем координаты x, y определяют пространственное положение на плоскости, а координата z – положение по высоте. С помощью плоскостных координат x и y адресуется некая «абстрактная клетка» на «монтажном поле», которая похожа на шахматную доску. Надо учитывать тот факт, что деталь имеет геометрические размеры и протяженность, поэтому надо как-то определять положение всех частей детали. При этом можно выделить два подхода:

I) условно разбить деталь на «абстрактные клетки» и описать координаты каждой такой клетки детали;

II) описать какой-нибудь определенный угол детали (базовую клетку детали) и указать направление положения детали (север, юг, восток, запад).

Если использовать аналогию с компьютерной графикой, то первый способ можно сравнить с растровой графикой, а второй – с векторной.

в) смешанный «относительно-абсолютный». Этот способ является комбинацией двух предыдущих. В рамках этого способа можно выделить несколько разновидностей.

И) Способ с несколькими базовыми деталями. При этом выбирается несколько базовых деталей, задаются их абсолютные координаты и описание производится относительно этих базовых деталей. Пусть определены несколько базовых деталей b_1, b_2, \dots, b_n . Деталь X может описываться относительно любой из них. Кроме того, допускается описание одной и той же детали относительно разных базовых деталей. Однако, при этом данные описания должны быть согласованы и не противоречить одно другому.

П) Способ с относительной адресацией. В этом случае задаются не абсолютные координаты детали, а относительные (относительно базы), например $x+2, y-1, z+1$.

Б. Классификационный признак: Точки зрения

Подходы:

а) описание с точки зрения «Север»; б) описание с точки зрения «Юг»; в) описание с точки зрения «Восток»; г) описание с точки зрения «Запад».

В. Классификационный признак: Учет типов деталей.

Атрибутами деталей являются форма, размер и цвет. Совокупность этих атрибутов определяет тип деталей.

Подходы:

а) использование только однотипных деталей. Это упрощенный подход. Например, могут использоваться детали в виде прямоугольных параллелепипедов одного и того же размера и одного и того же цвета («кирпичики»). Как вариант, типоразмер может быть одинаковым, а цвет разным.

б) использование разнотипных деталей. Это универсальный подход. Таким образом, деталь определяется как экземпляр определенного типа, например, $d7:T1$.

Практически всякое описание LEGO-конструкций основывается на использовании пространственных отношений между деталями LEGO (а также между частями конструкций). Для представления LEGO-изделия на основе пространственных отношений удобно использовать графы. Как известно, графы идеально подходят для визуального представления бинарных отношений, к классу которых относятся и пространственные отношения.

Для уменьшения объема описания LEGO-изделия можно вводить и использовать *параметризованные пространственные отношения*. Например, если взять две квадратные пластины разных размеров – пластину $d1$ большого размера и пластину $d2$ малого размера, то можно составить множество пространственных отношений между ними, иными словами, множество взаимных расположений одной пластины относительно другой пластины. Пластина $d2$ может лежать на пластине $d1$, причем с разными смещениями относительно координат x и y . Отрицательные смещения можно интерпретировать как ситуацию нависания пластины $d2$ над пластиной $d1$. Каждое новое «смещение» приводит к новому пространственному отношению. Чтобы кардинально сократить размер описания можно ввести понятие параметризованного пространственного отношения. Например, для отношения «выше» для двух квадратных пластин можно ввести предикат вида $r99(d1, d2, x, y)$, где $d1$ $d2$ – это идентификаторы нижней и верхней пластины, соответственно, x и y – координаты пластины $d2$ в координатном поле пластины $d1$, причем за начало координат можно принять левый нижний угол пластины $d1$.

Первой задачей, предваряющей задачи преобразования, является задача проверки корректности описания. Описание должно соответствовать определенным требованиям корректности, непротиворечивости, целостности и согласованности.

Задачи преобразований представлены ниже.

а) Изменение базовой детали. Пусть имеется некоторое Prolog-описание 3D-конструкции при использовании «относительного» подхода, где базовой деталью является деталь X . Необходимо преобразовать данное описание для случая, когда базовой деталью станет деталь Y .

б) Изменение точки зрения. Имеется некоторое Prolog-описание 3D-конструкции при использовании точки зрения X . Необходимо преобразовать данное описание для случая, когда новой точкой зрения станет Y .

в) Изменение формы описания. Имеется некоторое Prolog-описание 3D-конструкции, которое было получено при использовании подхода X. Необходимо преобразовать данное описание для случая, когда используется другой подход Y. Например, 3D-изделие было описано при использовании «относительного» подхода, а надо преобразовать это к описанию в соответствии с «абсолютным» подходом (и наоборот).

г) Устранение избыточности в описаниях. Основным требованием к описанию 3D-конструкций является то, что это описание должно быть полным (т.е. полностью описывать конструкцию) и целостным (то есть удовлетворять ряду ограничений). В то же время описание может содержать избыточность. Пусть, например, описывается, что деталь X находится выше детали Y, а деталь Y находится ниже детали X. Очевидно, что одно из описаний является лишним и его можно удалить. Удаление описания не должно приводить к потере полноты и целостности описания. Описание, в котором избыточность полностью отсутствует, назовем минимальным.

д) Введение полной избыточности в описания. Необходимо дополнить имеющееся описание 3D-конструкции до наиболее полного описания, то есть такого описания, которое уже нельзя более никак дополнить. Назовем такое описание максимальным.

е) Перечисление всех возможных минимальных описаний. Одну и ту же 3D-конструкцию, как правило, можно описать разными способами. Таким образом, описание, как правило, не является однозначным. Встает задача получения всех возможных минимальных описаний.

Одним из направлений дальнейших исследований является использование принципов и методов работы 3D-принтера для описания и преобразований 3D-представлений LEGO-конструкций.

Библиографический список

1. Википедия. Четвёртая промышленная революция. – URL: https://ru.wikipedia.org/wiki/Четвёртая_промышленная_революция/
2. LEGO Digital Designer. – URL: <https://www.lego.com/en-us/ldd/download>
3. Сенокосов, И. В. Онтологическое и графовое представления 3D-конструкций из LEGO-блоков / И. В. Сенокосов, А. С. Войнов, В. Н. Дубинин // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. ст. V Ежегодной межвуз. науч.-практ. конф. (МНПК – 2018). – Пенза, 2018. – С. 1–4.
4. SWI Prolog. – URL: <http://www.swi-prolog.org/>

К. Н. Савинов, Д. А. Данилин, Р. А. Торопкин

Пензенский государственный университет, Пенза, Россия

АНАЛИЗ СОВРЕМЕННЫХ РЕШЕНИЙ ДЛЯ РАЗРАБОТКИ КРОСС-ПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Показаны результаты анализа наиболее популярных современных решений в области разработки кросс-платформенных приложений для мобильных устройств и их особенности.

Ежегодный рост числа пользователей мобильных устройств, а также присутствие на рынке нескольких популярных мобильных операционных систем (Android – 77 % устройств, iOS – 21 %, Windows Phone – 0,5 % и др.) способствуют положительной динамике спроса на разработку кроссплатформенного мобильного ПО [1]. В общем случае, задача разработки приложений для нескольких целевых мобильных ОС может решаться двумя способами – созданием отдельных нативных решений для каждой системы или использование средств кроссплатформенной разработки. Каждый из подходов различается между собой в возможностях реализации определенного функционала, сложности разработки, ее

длительности и стоимости. Данная статья освещает особенности каждого из этих подходов, а также приводит данные о существующих фреймворках кроссплатформенной разработки, их плюсах и минусах.

Ключевой особенностью кроссплатформенных приложений является то, что их структура и логика работы основывается на базе используемого фреймворка, а затем обрабатывается в нативные запускающие элементы, т.е. интегрируется в базовые проекты целевых ОС. Плюсами данного подхода можно считать экономию времени и ресурсов на создание ПО, гибкость и оперативность изменения бизнес-логики приложения, теоретическое уменьшение количества возможных ошибок разработки. Главными же недостатками являются более медленная (в сравнении с нативными приложениями) работа приложений, а также ограниченный инструментарий средств разработки, не позволяющий использовать уникальные особенности различных мобильных платформ.

Под нативной разработкой подразумевается использование оригинальных языков и инструментов разработки мобильной операционной системы (IDE XCode вкупе с языками программирования Objective-C, Swift, C и C++ для iOS; IDE Android Studio, Eclipse и язык программирования Java – для Android). Плюсами данного подхода можно считать широкие возможности IDE и хорошую оптимизацию приложений, высокую эффективность сборки готовых пакетов ПО. Минусами же данного подхода являются более высокая стоимость и увеличенные сроки разработки, необходимость унификации логики работы версий приложения для разных ОС и др. [2].

В отличие от немногочисленных нативных решений (по одному – два для каждой мобильной ОС), существует целый ряд фреймворков кроссплатформенной разработки, основанных на различных языках программирования. Рассмотрим некоторые из них.

Так, например, технология PhoneGap позволяет создавать мобильные приложения, используя классические языки web-разработки (HTML, CSS и JS), что обеспечивает низкий порог вхождения в процесс разработки на базе данной технологии для frontend-разработчиков [3]. PhoneGap Build позволяет компилировать сборки для iOS, Android и Windows Phone одновременно без необходимости установки дополнительных пакетов и библиотек, что уменьшает размер установочных файлов. Кроме того, в отличие от многих других решений, данный фреймворк позволяет осуществлять сборки проектов для iOS без использования компьютеров компании Apple.

Фреймворк Xamarin платформы .Net позволяет создавать проекты на языке программирования C# и языке описания интерфейса XAML с их последующей сборкой на различные платформы [4]. Для каждой доступной ОС в фреймворке существует суб-платформа со своими специфическими методами и классами, однако зачастую возможности этих платформ сильно урезаны в целях унификации интерфейса приложений – например, в них отсутствует стандартный элемент пользовательского интерфейса – «чекбоксы». Плюсом данного решения является его поддержка мощнейшей средой разработки Visual Studio, а также наличие подробной многоязычной документации (MSDN, Microsoft DOCS).

Среда разработки 2D и 3D приложений и игр Unity является одним из самых популярных и доступных инструментов для создания и обработки графического контента. Данное решение применяется как правило при разработке игр и позволяет создавать приложения, поддерживаемые не только мобильными, но и другими операционными системами – Windows, OS X, Linux, а также игровыми консолями Wii, PlayStation и Xbox. Минусом же данного решения является низкий уровень документированности среды разработки и сложность ее освоения (обилие различных настроек объектов). Кроме того, приложения, разработанные на Unity отличаются высокими затратами вычислительной мощности и памяти мобильного устройства.

Фреймворк Qt языка программирования C++ отличается наиболее широкими возможностями, предоставляемыми разработчику, так как изначально не ориентирована на конкретную область задач (Unity – разработка игр) или мобильные платформы (Xamarin). Qt стоит рассматривать не столько как набор классов для создания GUI, а скорее, как полноценный инструментарий. Платой за широкие возможности является тяжеловесность языка программирования и самого фреймворка для освоения.

React Native на данный момент является перспективной, молодой технологией, обеспечивающей ей с одной стороны популярность (в том числе ввиду базирования на web-технологиях), и наличие недоработок – с другой. Кроме того, отрицательно сказывается меньшее число визуальных компонентов для платформы Android (ее поддержка была добавлена позднее iOS). Данное решение привлекательно для проектов, в которых мобильные приложения являются адаптацией функционала web-версии сервиса, написанного на ReactJS, так как в этом случае скорость разработки возрастает в разы.

Cordova – платформа, позволяющая разрабатывать мобильные приложения для разных операционных систем путем встраивания браузера в мобильное приложение [5]. Таким образом, разрабатываемое приложение представляет собой мини-браузер, отображающий созданный контент. Плюсом такого подхода является отсутствие необходимости упаковки ресурсов для работы приложения в установочный файл – их можно загружать с сервера при необходимости. По умолчанию Cordova предоставляет только базовые возможности браузера, которые есть на данном мобильном устройстве, но позволяет расширять набор функций, доступных в браузере путем использования плагинов. Каждый плагин предоставляет унифицированный интерфейс, который можно использовать из браузера на разных платформах. Минусом такого решения является снижение скорости работы ввиду использования стандартного обозревателя страниц.

Как было сказано выше, и нативная, и кроссплатформенная разработка мобильных приложений имеют свои достоинства и недостатки. Главные достоинства нативных приложений – быстрое действие и использование всех возможностей каждой из платформ. Однако, они уступают кроссплатформенным решениям в задачах оперативной разработки и прототипирования ПО, так необходимого для обеспечения жизнеспособности продукта в динамичных условиях современного рынка. И несмотря на все большее распространение web-технологий во всех сферах, в том числе в разработке мобильных приложений, расширение их возможностей в сторону кроссплатформенности, вопрос о выборе того или иного инструмента создания продуктов для мобильных устройств еще долго будет оставаться открытым [6].

Библиографический список

1. Таненбаум, Э. Современные операционные системы / Э. Таненбаум, Х. Бос. – 4-е изд. – СПб. : Питер, 2017. – 1120 с.
2. Вячеслав, С. Мобильное приложение как инструмент бизнеса / С. Вячеслав. – М. : Альпина Паблишер, 2017. – 240 с.
3. Рассел, Дж. PhoneGap / Дж. Рассел, Р. Кох. – М. : Книга по Требованию, 2013. – 104 с.
4. Прайс, М. C# 7 и .Net Core. Кросс-платформенная разработка для профессионалов / М. Прайс. – СПб. : Питер, 2018. – 640 с.
5. Стайер, Р. Cordova / Р. Стайер. – М. : Книга по Требованию, 2017. – 412 с.
6. Тузовский, А. Проектирование и разработка web-приложений : учеб. пособие / А. Тузовский. – М. : Юрайт, 2017. – 218 с.

М. А. Барышев, Л. В. Гурьянов

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ КЛАСТЕРНОГО АНАЛИЗА ДЛЯ РЕШЕНИЯ ЗАДАЧ МАРКЕТИНГА НА ПРЕДПРИЯТИИ

Рассмотрены постановка задачи классификации клиентов на основе кластерного анализа для повышения эффективности маркетинговой деятельности предприятия.

Методы кластерного анализа широко применяются как в научных исследованиях, так и задачах практического характера.

Кластерный анализ (англ. *cluster analysis*) – статистическая многомерная процедура, которая выполняет сбор данных, которые содержат информацию о выборе объектов, и потом упорядочивающая объекты в однородные сравнительно группы.

Кластерный анализ – это разбиение множества объектов на группы (кластеры), основываясь на свойствах этих объектов. В общем случае кластер-анализ предназначен для объединения некоторых объектов в классы (кластеры) таким образом, чтобы в один класс попадали максимально схожие, а объекты различных классов максимально отличались друг от друга. Количественный показатель сходства рассчитывается заданным способом на основании данных, характеризующих объекты [1].

Пусть множество $I = \{I_1, I_2, \dots, I_n\}$ обозначает n объектов некоторой совокупности. Существует некоторое множество показателей $C = (C_1, C_2, \dots, C_k)^T$, которыми обладает каждый объект. Для множества I известно множество $X = \{X_1, X_2, \dots, X_n\}$, описывающее множество I , где X_j – вектор показателей множества C для j -го объекта. Множество X можно представить как n точек в k -мерном евклидовом пространстве.

Общая постановка задачи кластерного анализа имеет следующее представление. Необходимо на основании данных, содержащихся в множестве X , разбить множество объектов I на m кластеров, так чтобы каждый объект множества I принадлежал одному и только одному подмножеству разбиений, и чтобы объекты, принадлежащие одному и тому же кластеру, были сходными, в то время как объекты, принадлежащие разным кластерам, были разнородными.

В кластерном анализе для количественной оценки сходства вводится понятие «расстояние между объектами». Чаще других для объектов, характеризующихся числовыми признаками, используется формула евклидова расстояния:

$$p(x_i, y_j) = \sqrt{\sum_{m=1}^k (x_{im} - x_{jm})^2}, \quad (1)$$

где $x_i = (x_{i1}, \dots, x_{ik})$ и $x_j = (x_{j1}, \dots, x_{jk})$ – векторы, составленные из значений k признаков i -го и j -го объектов соответственно [2].

Рассмотрим конкретную задачу разделения клиентов некоторой организации на категории с целью построения наиболее эффективной стратегии взаимодействия с ними и получения наибольшей прибыли.

За множество X принимаем множество данных о сделках рассматриваемой организации, за I – множество клиентов этой организации. Разбиение на кластеры производится на основании следующих параметров:

- количество сделок с клиентом;
- количество приобретенного товара;
- сумма сделок;
- и ряд других.

Клиентов необходимо разделить на заданное количество категорий (VIP-клиенты, клиенты категории А, категории В и т.д.). Для решения задачи кластеризации используем вычисление расстояния между объектами по формуле 1.

Программное обеспечение для решения задач кластеризации разрабатывается на платформе «1С Предприятие».

Результат кластеризации может быть выведен в дендрограмму – специальный объект, предназначенный для отображения последовательных связей между объектами (рис. 1).

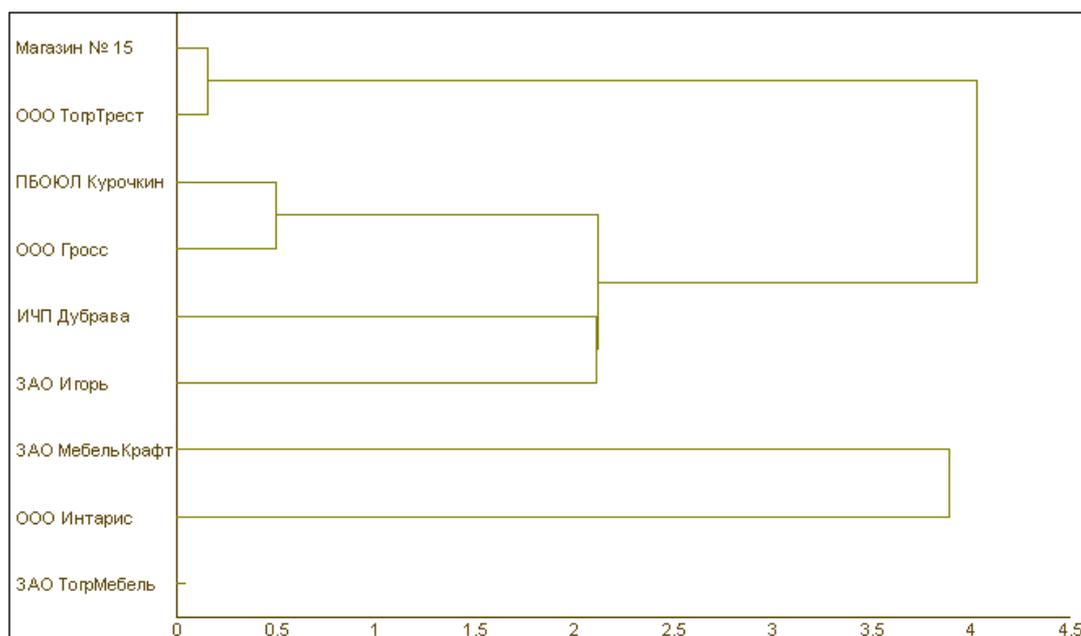


Рис. 1. Дендрограмма связей

Использование кластерного анализа обеспечивает достижение следующих целей при обработке сложных производственных данных:

- упрощение обработки за счет применения ко всем объектам кластера одного и того же метода анализа данных;
- в случае большой исходной выборки её можно сократить, оставив от каждого кластера по одному самому типичному представителю, что в конечном итоге приведет к существенному сжатию данных;
- обнаружение новизны в исходных данных в случае, если находятся «нетипичные» объекты, которые не принадлежат ни к одному из выделенных кластеров.

Библиографический список

1. Сенкальский, Р. А. Кластерный анализ разработки современных алгоритмов обработки данных / Р. А. Сенкальский, С. А. Страмоусова // Молодой ученый. – 2016. – № 9.4. – С. 63–65.
2. Дюран, Б. Кластерный анализ / Б. Дюран, П. Оделл // Статистика. – М., 1977. – С. 11–19.

И. А. Долгова, И. П. Грибов

Пензенский государственный университет, Пенза, Россия

ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ БЛОКЧЕЙН В ЭКОНОМИКЕ

Рассмотрены перспективы применения технологии блокчейн в экономике. Отмечены преимущества и недостатки применения системы распределённого реестра. Рассмотрены пути решения проблем, возникающих при использовании технологии блокчейн в сфере цифровой экономики.

Появление системы блокчейн связано с появлением криптовалюты биткоина. В 2008 году была опубликована статья, посвященная цифровой пиринговой системе платежей, авторство которой приписывается человеку или нескольким людям под псевдонимом Сатоши

Накамото. В данной статье была рассмотрена проблема того, как совершать денежные транзакции, минуя посредников. Для решения этой проблемы ранее предлагалось использовать электронную подпись с целью удостоверения личности. Но такое не всегда возможно, поэтому в качестве решения предлагались варианты с использованием пиринговой системы между участниками транзакций. Этот метод получил название блокчейн.

Блокчейн – это специальная многоэтапная технология, предназначена для надежного хранения данных и учёта транзакций. Эта система представляет собой цепочку блоков, соединённых между собой. Блоки представляют собой базы данных, объём которых постоянно растёт. Каждый следующий блок содержит информацию о предыдущих транзакциях. Как только блок сформировался, происходит сравнение данных с предыдущими блоками. После создания блока его редактирование невозможно. Копии, которые хранятся на других компьютерах, автоматически обновляются. Следует отметить, что блоки хранятся в виде определённого списка, а участники сети представляют собой узлы, которые содержат в себе массивы данных. Узлы могут добавлять необходимую информацию в конец списка и согласовывать изменения в сети.

Алгоритм работы системы распределенного реестра, функционирующей согласно технологии блокчейн, выглядит следующим образом:

1. Создаётся новая независимая транзакция, которая автоматически отсылается всем узлам сети, входящим в состав системы. Сеть участников работает по принципу одноранговой, децентрализованной пиринговой сети. Затем начинается процесс обработки новой информации.

2. Специальные компьютеры добавляют новую транзакцию в блок.

3. Определённый майнер, (специальная программа, которая подбирает код), пытается подобрать специальный хеш – блок (ключ), для активации записи в блок. Данный процесс получил название proofofstake.

4. После подбора ключа данные передаются всем участникам системы.

5. Передаваемая информация проверяется на корректность и отсутствие повторения каждым участником системы. Если данный блок уже существует и информация повторяется, то запись в блок не осуществляется [1].

Следует отметить, что вся информация, записанная в блок, подтверждается криптографической защитой. Каждый пользователь при установке системы блокчейн на свой персональный компьютер получает свои индивидуальные ключи: первый ключ используется для шифрования транзакций, второй для верификации транзакций. Добавляя транзакцию, участник оставляет свой хеш (подпись). Благодаря этому система предоставляет возможность просмотра всех текущих и выполненных транзакций.

Рассмотрим преимущества и недостатки технологии блокчейн.

Самым важным преимуществом данной технологии является децентрализация. Именно она позволяет совершать финансовые операции не через один узел, а через всю систему, делая её прозрачной. Примером является система расчетов – Ripple [2]. Другим не менее важным достоинством является оплата без посредников на основе смарт – контракта. Кроме того, системы на основе технологии блокчейн обеспечивают повышенную безопасность хранящихся в них данных. Данную систему невозможно взломать, так как копии хранятся на многих компьютерах. В систему может производиться только запись блоков, а изменение внесённой информации невозможно.

Недостатком данной системы является большая стоимость оборудования для установки технологии блокчейн.

В настоящее время в различных государствах проводятся исследования по расширению области использования технологии блокчейн.

Например, в Великобритании государственным управлением науки (distributed ledger technology beyond blockchain) проводятся исследования с целью изучения возможности использования системы распределенного реестра в сфере бизнеса и управления процессом оказания государственных услуг гражданам. При этом государство должно выступать в роли инициатора и заказчика одновременно, а граждане должны использовать смарт – контракты (SmartContracts). Результаты исследований показали, что применение тех-

Аппаратно-программное обеспечение информационно-вычислительных систем и систем управления

нологии блокчейн позволит улучшить контроль выполнения бизнес-процессов в государственном управлении [3].

В 2016 г. в США представители Совета Федерации резервной системы и Федерального резерва банков Нью-Йорка и Чикаго опубликовали статью, в которой описывалось применение технологии блокчейн в экономике США [4].

Использование данной технологии в экономической сфере предоставляет следующие преимущества:

- применение смарт – контракта позволит ускорить процесс юридического и экономического управления сделками, отследить необходимые документы, уменьшить вероятность ошибки, сократить расходы на оформление контрактов;
- уменьшение влияния банков, которые выступают в качестве посредников при заключении сделок.

В то же время при внедрении новой технологии блокчейн возникают две основные проблемы:

- 1) для генерации блоков необходимо специализированное оборудование с высоким энергопотреблением;
- 2) для хранения информации опроверждённых транзакций требуются значительные объёмы дисковой памяти.

Для уменьшения энергопотребления, что резко снижает стоимость затрат на генерацию блоков, в настоящее время используются специальные среды программирования. Так, например, в системе биткойна время генерации составляет 10 минут, а в системе Эфириум данное действие занимает 12 секунд [5]. Ещё одним методом решения данной проблемы является использование технологии Lightning Network, в которой транзакции отправляются напрямую участникам, а в блокчейне существуют только «контрольные точки».

Для уменьшения объема потребляемой памяти и сведения информации в транзакции к минимуму предлагается хранить не всю информацию о сгенерированных блоках, а только их заголовки, описывающие транзакции. Именно этот метод (Rollerchain) применяется на блокчейн – платформе Scortex [6].

Таким образом, использование технологии блокчейн в экономике является перспективным направлением её развития. В настоящее время наблюдается всё больший интерес государственных и негосударственных экономических структур к использованию технологии блокчейн и криптовалюты в области цифровой экономики. Например, в Японии используют блокчейн как основную платформу банкинга [7]. В России в 2017 году на совещании с премьер – министром Дмитрием Медведевым было озвучено развитие и применение технологии блокчейн в сфере экономики.

Библиографический список

1. Nakamoto, S. A Peer-to-Peer Electronic Cash System / S. Nakamoto // Bitcoin. – URL: <https://bitcoin.org/bitcoin.pdf>
2. Криптовалюта Ripple. – URL: <https://tehnoobzor.com/cryptolife/ripple/2249-chem-otlichaetsya-bitkoin-ot-ripple.html>
3. Ledger Technology: beyond block chain. A report by the UK Government Chief Scientific Adviser // Government Office for Science, 2016. – URL: https://www.gov.uk/government/news/distributed-ledger-technology-beyond-block-chain_data/file/492972/gs-16-1-distributed-ledger-technology.pdf
4. Distributed ledger technology in payments, clearing, and settlement//Finance and Economics Discussion Series 2016-095. – Washington : Board of Governors of the Federal Reserve System, 2016. – P. 34. – URL: <https://www.federalreserve.gov/econresdata/feds/2016/files/2016095pap.pdf>
5. Соловьев, А. Блокчейн: подводные камни / А. Соловьев // Открытые системы. СУБД. – 2016. – № 4. – URL: <http://www.osp.ru/os/2016/04/13050987/>
6. Пряников, М. М. Блокчейн как коммуникационная основа формирования цифровой экономики: преимущества и проблемы / М. М. Пряников, А. В. Чугунов. – URL: <http://injoit.org/index.php/j1/article/viewFile/447/499>
7. Совещание с вице-премьерами // Правительство России [06.03.2017]. – URL: <http://government.ru/news/26650/>

Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ БАГ-ТРЕКИНГОВЫХ СИСТЕМ ДЛЯ ТЕСТИРОВАНИЯ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

В настоящее время в различных сферах деятельности человека, например, таких как: нефтегазовой, банковской, обслуживания клиентов операторов сотовой связи, широкое применение получили автоматизированные системы. Особую сложность при разработке и эксплуатации таких систем представляют автоматизированные системы реального времени.

Автоматизированные системы реального времени на этапе сопровождения часто требуют внесения различных изменений, адаптации к новым условиям работы. Основной проблемой при доработке и адаптации таких систем является невозможность прекращения деятельности системы. Вследствие этого все изменения, в том числе в программном обеспечении, необходимо вносить в режиме онлайн. Таким образом, сложной задачей является корректировка программного обеспечения автоматизированных систем без приостановления их работы, которая требует проведение тестирования всей системы, так как внесение изменений в режиме онлайн может повлечь нарушения в работе составляющих ее подсистем.

Чаще всего для решения этой задачи в таких системах применяется автоматизированное тестирование.

Для проведения автоматизированного тестирования таких систем используются специальные программные средства, так называемые баг-трекинг-системы или системы внесения ошибок. Система отслеживания ошибок – прикладная программа, разработанная с целью помочь разработчикам программного обеспечения (программистам, тестировщикам и др.) учитывать и контролировать ошибки и неполадки, найденные в программах, пожелания пользователей, а также следить за процессом устранения этих ошибок и выполнения или невыполнения пожеланий [1].

Многие компании находят, что интегрированная система отслеживания ошибок уменьшает время простоя, увеличивает производительность и повышает удовлетворенность клиента от работы с их системами. Вместе с полным раскрытием информации, открытая система отслеживания ошибок позволяет производителям программного обеспечения поддерживать контакт со своими клиентами, для передачи сообщений об ошибках по всей цепи управления данными. Многие корпорации также обнаруживают, что отслеживание ошибок уменьшает издержки, службы поддержки ИТ [4].

Главный компонент баг-трекинг-систем – база данных, содержащая сведения об обнаруженных дефектах. Эти сведения могут включать в себя: номер (идентификатор) дефекта, короткое описание дефекта, кто сообщил о дефекте, дата и время, когда был обнаружен дефект, версия продукта, в которой обнаружен дефект, серьезность (критичность) дефекта и приоритет решения, описание шагов для выявления дефекта (воспроизведения непреднамеренного поведения программы), ожидаемый результат и фактический результат, кто ответственен за устранение дефекта, обсуждение возможных решений и их последствий, текущее состояние (статус) дефекта, версия продукта, в которой дефект исправлен [1].

Для тестирования сложных систем необходимо выбрать соответствующую баг-трекинг-систему. В качестве критериев выбора предлагается использовать следующие параметры:

- конфигурация (настройка полей на формах программного средства);
- возможность фильтрации данных в таблицах выборки;
- разграничение доступа по средствам назначения прав пользователям;

- стоимость лицензии;
- язык интерфейса;
- легкость восприятия интерфейса для неопытного пользователя.

В настоящее время наиболее распространены следующие баг-трекинг-системы:

а) Jira – коммерческая веб-система отслеживания ошибок, разработанная компанией Atlassian [2].

Система основана на Java. Основной элемент учёта в системе – задача. Задача содержит название проекта, тему, тип, приоритет, компоненты и содержание. Задача может быть расширена дополнительными полями (также и новые пользовательские поля могут быть определены), приложениями (например – фотографиями, скриншотами) или комментариями. Любые изменения в задаче протоколируются в журнал.

Jira имеет большое количество возможностей конфигурации: для каждого приложения может быть определен отдельный тип задачи с собственным workflow, набором статусов, одним или несколькими видами представления (англ. screens). Имеет поиск по критериям и фильтрацию. Кроме того, с помощью так называемых «схем» можно определить для каждого индивидуального Jira-проекта собственные права доступа, поведение и видимость полей и многое другое. Обладает дружественным интерфейсом.

Для интеграции с внешними системами Jira поддерживает интерфейсы SOAP, XML-RPC и REST. Поставляется со средствами интеграции с такими системами управления версиями, как Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial и Perforce. Существуют дополнения, позволяющие встроить Jira в интегрированные среды разработки, в том числе Eclipse и IntelliJ IDEA. Переведена на многие языки, включая русский, английский, японский, немецкий, французский, испанский [3].

Jira является коммерческим продуктом. Стоимость лицензии определяется количеством пользователей (в среднем \$10 за пользователя) [2].

б) Bugzilla – свободная распространяемая система отслеживания ошибок с веб-интерфейсом на английском языке.

В 1998 году Bugzilla была выпущена как открытое программное обеспечение компанией Netscape. По состоянию на 2012 год разрабатывается фондом Mozilla Foundation [4].

Первоначально Bugzilla была реализована на языке Tcl. В настоящее время Bugzilla 2.0 реализована на языке Perl [5].

Bugzilla имеет разделение по правам доступа. В системе реализована функция фильтрации. Так же Bugzilla не обладает интуитивным интерфейсом и возможностью настраивать поля.

в) Redmine – свободно распространяемое серверное веб-приложение для управления проектами и задачами (в том числе для отслеживания ошибок). Redmine написан на Ruby и представляет собой приложение на основе веб-фреймворка Ruby on Rails/5/.

Данный продукт предоставляет следующие возможности: ведение нескольких проектов, понятный и доступный интуитивный интерфейс, гибкая система доступа, основанная на ролях, система отслеживания ошибок, разграничение прав доступа, построения диаграммы Ганта и календарь, ведение новостей проекта, документов и управление файлами, оповещение об изменениях с помощью RSS-потоков и электронной почты, учёт временных затрат, настраиваемые произвольные поля для инцидентов, временных затрат, проектов и пользователей, лёгкая интеграция с системами управления версиями (SVN, CVS, Git, Mercurial, Bazaar и Darcs), создание записей об ошибках на основе полученных писем, возможность самостоятельной регистрации новых пользователей, многоязычный интерфейс (в том числе русский), поддержка СУБД MySQL, Microsoft SQL Server, PostgreSQL, SQLite, Oracle [5].

г) YouTrack – коммерческая англоязычная система отслеживания ошибок, программное обеспечение для управления проектами, разработанное компанией JetBrains. YouTrack поддерживает поисковые запросы, автодополнение, манипуляцию с наборами задач, настройку набора атрибутов задачи, создание пользовательских рабочих процессов и реализует подход, основанный на преимущественном использовании клавиатуры.

YouTrack разработан в соответствии с парадигмой языково-ориентированного программирования, использует язык JetBrains MPS и набор собственных предметно-ориенти-

рованных языков программирования, разработанных компанией JetBrains. Система использует встроенную базу данных Jet Brains Database – транзакционное хранилище пар «ключ – значение». Для удалённых вызовов процедур использует REST-стиль [7].

Пользовательский интерфейс YouTrack основан на технологии AJAX и позволяет производить управление как с помощью клавиатуры, так и с помощью мыши. Поиск и фильтрация задач осуществляется с помощью поисковых запросов, выполняемых в единой строке поиска.

YouTrack REST ful API позволяет разработчикам производить различные действия программно, в том числе:

- импорт существующих проектов и задач из других систем отслеживания ошибок;
- создание, изменение, получение всех атрибутов задач;
- управление проектами, пользователями, группами и ролями;
- возможность настройки конфигурации полей [7].

YouTrack доступен также в виде сервиса (SaaS), под названием YouTrack In Cloud, бесплатно для 10 пользователей и для 25 пользователей – 500 долларов в год, 50 пользователей – 750 долларов в год [8].

К недостаткам системы YouTrack можно отнести сложную нумерацию, так как номер задачи меняется при переносе ее в другой проект, а так же минусом является отсутствие техподдержки.

д) Team Foundation Server (сокр. TFS) – продукт корпорации Microsoft, представляющий собой комплексное решение, объединяющее в себе систему управления версиями, сбор данных, построение отчётов, отслеживание статусов и изменений по проекту и предназначенное для совместной работы над проектами по разработке программного обеспечения. Данный продукт доступен как в виде отдельного приложения, так и в виде серверной платформы для VisualStudioTeamSystemс доступным для понимания интерфейсом и есть на русском языке [9].

Большая часть действий в Team Foundation Server происходит с «рабочими элементами», то есть тест кейсами. Рабочими элементами называются отдельные единицы (шаги) работы, выполняемые поочерёдно. Во многих источниках они отождествляются с элементами типа «ошибка» (bug) в системах отслеживания ошибок наподобие Bugzilla, т.е. в этом случае рабочий элемент имеет поля Area (связанная область), Iteration (состояние), Assignee (связан с), Reported By (кем создан) для указания соответствующей информации, историю, приложенные файлы и другие настраиваемые атрибуты. Каждый рабочий элемент имеет соответствующие методики контроля, которые определяют, кому какие именно элементы доступны и какие действия он с ними может производить (просмотр, фильтрация, редактирование, создание, удаление и др.) [10].

В среднем стоимость лицензии системы TFS составляет около 30 000 рублей, в зависимости от параметров заказа [11].

Таким образом, в соответствии с выбранными критериями подбора для тестирования автоматизированных систем, имеющих сложную структуру, настройку и модернизацию которых, необходимо проходить без остановки работы системы, наиболее подходящей баг-трекинговой системой для России является Team Foundation Server, так как данная система обладает интуитивным и дружелюбным интерфейсом на русском языке, позволяет видоизменять поля на страницах системы для удобства пользователей и имеет доступную для компаний стоимость лицензии.

Библиографический список

1. Система отслеживания ошибок // Википедия. – URL: https://ru.wikipedia.org/wiki/Система_отслеживания_ошибок
2. Jira // ATLASSIAN. –URL: <https://ru.atlassian.com/software/jira/pricing>
3. Система управления проектами и задачами JIRA компании Atlassian и ее применение // ЛионСофт. – URL: <http://jira.ru/jira.shtml>
4. Краткое введение в Bugzilla // Mozilla в России. – URL: <http://mozilla-russia.org/products/bugzilla/bugzilla.html>
5. Redmine // Redmine – лучшие решения. – URL: <https://www.easyredmine.com/ru/>

6. Оперативное планирование в Redmine // Habr. – URL: <https://habr.com/post/245065/>
7. YouTrack – онлайн баг-трекер и инструмент для управления задачами // Продукты и разработки. – URL: <https://jetbrains.ru/products/youtrack/>
8. Сравнение сервисов JIRA и YouTrack // Startpack. – URL: <https://startpack.ru/compare/jira/youtrack>
9. TeamFoundationServer // Википедия. – URL: https://ru.wikipedia.org/wiki/Team_Foundation_Server
10. Архитектура TeamFoundationServer // Microsoft. – URL: [https://msdn.microsoft.com/ru-ru/library/ms252473\(v=vs.120\).aspx](https://msdn.microsoft.com/ru-ru/library/ms252473(v=vs.120).aspx)
11. Работа с TeamFoundationServer 2010 // Habr. – URL: <https://habr.com/post/90911/>

О. С. Дорофеева, Б. В. Казаков, И. А. Казакова

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА К РАЗРАБОТКЕ И МОДЕЛИРОВАНИЮ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ

Рассматриваются вопросы применения объектно-ориентированного подхода при разработке и моделировании сложных программных систем.

При объектно-ориентированном подходе к разработке и моделированию сложных программных систем каждый объект системы является представителем некоторого класса однотипных объектов. Класс определяет общие свойства для всех его объектов и методы, определяющие взаимодействие объектов этого класса с внешней средой. Таким образом, объектно-ориентированный подход к разработке и моделированию сложных программных систем основан на тесной взаимосвязи данных и методов их обработки. Объектно-ориентированный подход к разработке и моделированию сложных программных систем позволяет представить программную систему в виде совокупности классов и объектов. При этом программная система представляется в виде иерархии классов, а ее функционирование рассматривается как взаимодействие объектов. Такой подход позволяет описать систему наиболее естественным образом.

Применение объектно-ориентированного подхода к разработке и моделированию сложных программных систем имеет ряд преимуществ:

1. Повторное использование классов. Разрабатываемые классы отражают типовые проектные решения, поэтому их использование возможно и в других проектах. Объектно-ориентированный подход позволяет накапливать опыт проектных решений в виде библиотек классов. При наличии развитых библиотек классов проектирование и программирование новых программных систем будет в основном сводиться к сборке системы из готовых фрагментов.

2. Упрощение внесения изменений. При объектно-ориентированном подходе изменения в проекте имеют локальный характер.

3. Распараллеливание работ. Программирование и тестирование отдельных компонент системы возможно до завершения проектирования программной системы в целом, что экономит время разработки.

Объектно-ориентированный подход к разработке и моделированию сложных программных систем содержит несколько этапов. Начальным этапом – объектно-ориентированный анализ. На этом этапе определяются требования к системе. Следующий этап – анализ предметной области. Здесь определяются основные классы и объекты программной системы.

Далее следует этап объектно-ориентированного проектирования, который детализирует представление классов и объектов, полученных на этапе анализа. Определяются структуры данных, методы, отношения между классами, разрабатываются сценарии взаи-

действия объектов. При проектировании системы могут вводиться новые классы и объекты. Результатом проектирования должна быть детальная спецификация объектов, классов и отношений, достаточная для их программирования.

Программирование, тестирование и сборка системы рассматриваются как единый этап. На этом этапе возможно введение новых классов, изменение структур данных, добавление новых методов. Программирование и тестирование отдельных компонент системы возможно до завершения проектирования, что экономит время разработки.

На этапе модификации системы активно используются свойства классов и объектов, которые позволяют легко локализовать изменения, свести к минимуму переработку программ за счет использования механизма наследования. В современных условиях в связи с повышением требований к программным системам, применением более сложных архитектурных решений и как следствие с увеличением объемов программного кода разработка программной системы путем прямого перехода от требований к программированию приводит к огромному количеству ошибок, в результате чего такие проекты обычно остаются незавершенными, поэтому создание современных программных систем невозможно без применения объектно-ориентированного подхода к разработке и моделированию сложных программных систем [1].

Модель программной системы создается для того, чтобы лучше понимать разрабатываемую систему. Проектирование программных систем начинается с построения модели, описывающей систему с разных точек зрения. Модели позволяют свести высокую сложность программных систем до уровня, понимаемого человеком. Достигается это за счет иерархического принципа их построения и применения наглядной графической нотации. Иерархия уровней описания системы дает возможность резко сократить количество элементов, которые должен анализировать человек. Каждая модель может быть выражена с разными уровнями детализации. При этом на верхних уровнях иерархии опускаются детали реализации, которые проявляются на более низких уровнях [2].

В процессе построения модели программных систем не требуется детальное кодирование. Таким образом, происходит разделение труда: моделирование программных систем выполняется обычно наиболее квалифицированными разработчиками, способными принять и обосновать кардинальные решения, а программную реализацию могут осуществлять программисты среднего уровня, воплощающие принятые и отраженные в модели системные решения в программных кодах. Построенные модели могут модифицироваться в течение всего жизненного цикла. Главное достоинство моделей – представление информации в наглядной графической форме. Для одинакового понимания графического представления разными людьми графическая нотация должна быть унифицирована, то есть должен использоваться язык, определяющий такую нотацию и ее семантику. Именно таким и является Unified Modeling Language (UML). С точки зрения объектно-ориентированного подхода к разработке и моделированию сложных программных систем достаточно полная модель сложной системы представляет собой определенное число взаимосвязанных представлений, каждое из которых адекватно отражает структуру программной системы. Сложность моделирования предметной области и разработки сложных программных систем привело к появлению новой методологии – объектно-ориентированный анализ и проектирование. Объектно-ориентированный анализ и проектирование – технология разработки программных систем, в основу которых положена объектно-ориентированная методология представления предметной области в виде объектов, являющихся экземплярами соответствующих классов [3].

Библиографический список

1. Орлов, С. А. Технологии разработки программного обеспечения : учеб. для вузов / С. А. Орлов. – 3-е изд. – СПб. : Питер, 2004. – 527 с.
2. Бозм, Б. У. Инженерное проектирование программного обеспечения : пер. с англ. под ред. А. А. Красиловой / Б. У. Бозм. – М. : Радио и связь, 1985. – 512 с.
3. Соммервилл, И. Инженерия программного обеспечения / И. Соммервилл. – 6-е изд. – М. : Вильямс, 2002. – 624 с.

О. С. Дорофеева, И. А. Казакова, Б. В. Казаков

Пензенский государственный университет, Пенза, Россия

ПРОГРАММНЫЕ СРЕДСТВА РАСПОЗНАВАНИЯ ЯЗЫКОВ И ГРАММАТИК

Посвящена обзору наиболее востребованных генераторов синтаксических анализаторов для формальных языков и грамматик.

При разработке финансовых, корпоративных и других программных систем может возникать необходимость в создании языка, специализированного на решении определенного круга задач. Такой язык называется предметно-ориентированным (англ. domain-specific language, DSL), например HTML – язык гипертекстовой разметки, Verilog и VHDL – языки описания аппаратуры, грамматики YACC и ANTLR для создания парсеров (синтаксических анализаторов). Предметно-ориентированные языки позволяют решать поставленные задачи в терминах и на уровне абстракции предметной области. Различают встроенные (embedded или internal) и внешние (external) DSL. Встроенные DSL реализуются в виде библиотек, использующих синтаксис некоторого языка общего назначения, но добавляющих в него элементы (типы данных, процедуры, методы), специфичные для данной предметной области [1]. Внешние DSL имеют собственную грамматику, а значит, требуют специфичных средств лексического и синтаксического анализа [1]. Следовательно, возможно разработать настолько выразительный внешний DSL, что специалисты предметной области смогут не только читать, но и писать код на этом языке. Для встроенного DSL это может оказаться трудно осуществимой задачей.

Для создания синтаксических анализаторов внешних предметно-ориентированных языков часто применяются генераторы парсеров. Примером может служить ANTLR (ANother Tool for Language Recognition) – генератор парсеров, который позволяет автоматически создавать программу-парсер (как и лексический анализатор) на одном из целевых языков программирования (C++, Java, C#, Python, Ruby) по описанию грамматики на языке, близком к расширенной форме Бэкуса-Наура (РБНФ) [2].

Преимущества: свободное программное обеспечение, использование единой нотации для описания лексических и синтаксических анализаторов, применение нисходящего, а не восходящего анализа, удобство работы с абстрактным синтаксическим деревом, предоставление сообщений об ошибках и восстановление после них. Также наличие визуальных сред разработки (ANTLR Works, ANTLR Studio, плагинов к Eclipse и IntelliJ IDEA) позволяют создавать и отлаживать грамматики, поддерживают подсветку синтаксиса, автодополнение, визуальное отображение грамматик, строящееся в реальном времени по мере ввода, отладчик, рефакторинг.

ANTLR широко используется в крупных известных продуктах:

- Реализация языков программирования Groovy, Jython, Processing, Apex
- Анализ языков запросов в системах баз данных Hibernate HQL, Cassandra, Hive, Pig
- Анализ языков программирования в средах разработки: NetBeans C++, Oracle SQL Developer IDE, IntelliJ IDEA Clion.

Yacc (Yet Another Compiler Compiler) – компьютерная программа, служащая стандартным генератором синтаксических анализаторов в Unix-системах. Yacc генерирует парсер на основе аналитической грамматики, описанной в нотации BNF (форма Бэкуса-Наура) или контекстно-свободной грамматики. На выходе yacc выдается код парсера на языке программирования Си.

Yacc был разработан Стефеном Джонсоном (Stephen C. Johnson) в AT&T для операционной системы Unix. Позже были написаны совместимые версии программы, такие как Berkeley Yacc, GNU bison, MKS yacc и Abxaxas yacc (обновлённый вариант AT&T-версии с

открытым исходным кодом также вошёл в проект OpenSolaris от Sun). Каждый вариант предлагал незначительные улучшения и дополнительные возможности по сравнению с оригиналом, но концепция осталась той же. Yacc также был переписан на других языках, включая Java, C#, Pascal, Ada, Ratfor, EFL, ML, Limbo, Erlang, Go и т.д.

Поскольку парсер, генерируемый с помощью yacc, требует использования лексического анализатора, то часто он используется совместно с генератором лексических анализаторов, в большинстве случаев это lex либо flex. Стандарт IEEE POSIX P1003.2 определяет как функциональность, так и требования для lex и yacc [3].

В качестве альтернативного подхода при построении синтаксических анализаторов можно моделировать парсеры как функции и определять функции высшего порядка (комбинаторы), снабженные грамматическими конструкциями, такими как упорядочение, выбор и повторение [4]. Данный подход применим для функциональных языков. При этом сложные парсеры строятся динамически из более простых посредством парсер-комбинаторов.

Для реализации подобного подхода при создании анализатора внешнего предметно-ориентированного языка можно использовать язык программирования Scala, использующий чистую объектно-ориентированную модель и полноценно поддерживающий функциональную парадигму программирования. Исходный код Scala может быть скомпилирован в байт-код, исполняемый виртуальной машиной Java. Таким образом, разрабатываемый на Scala анализатор может свободно взаимодействовать с Java-кодом. Это особенно полезно, если на момент начала разработки уже существует API, реализованное на языке Java и позволяющее решать задачи, специфичные для данной предметной области.

Библиотека Scala включает в себя программный каркас парсер-комбинаторов, позволяющий создавать анализаторы текстов на языках, которые могут быть описаны контекстно-свободной грамматикой. При этом главной привлекательной особенностью данного программного каркаса является предоставляемый им внутренний DSL: описание парсера на нем близко к аналогичному описанию с использованием РБНФ.

В качестве преимущества можно также отметить то, что программный каркас парсер-комбинаторов Scala проще в изучении и использовании, чем специализированные инструменты по генерации парсеров. При этом разница в производительности полученных парсеров часто оказывается незначительной, если только не требуется осуществлять разбор больших входных данных [4]. Кроме того, парсеры, реализованные средствами Scala, не имеют «тяжеловесных» зависимостей времени выполнения.

Таким образом, можно заключить, что если этап анализа текстов на разрабатываемом предметно-ориентированном языке не будет критичным по времени выполнения или подразумевается, что не будет осуществляться разбор больших объемов входных данных, то разработка парсера с использованием программного каркаса парсер-комбинаторов Scala может оказаться приемлемым решением.

Несмотря на это, на данный момент для реализации внешних DSL Scala используется редко. Более широкое распространение получили встроенные DSL на базе Scala (Pistache DSL, Apache Camel Scala DSL).

Следует отметить, что язык Scala динамично развивается, при этом, естественно, совершенствуется и программный каркас парсер-комбинаторов. Так, в версии библиотеки 2.8 была улучшена поддержка восстановления после ошибок и сообщений о них. Кроме того, была добавлена возможность создания так называемых *packrat*-парсеров. *Packrat*-парсер является разновидностью анализатора, работающего схожим с рекурсивным спуском методом, за исключением того, что при анализе он запоминает промежуточные результаты всех вызовов взаимно рекурсивных функций анализа. Благодаря этой особенности *packrat*-парсер способен анализировать множество контекстно-свободных грамматик и любую РВ-грамматику (грамматику, разбирающую выражение) в линейное время ценой увеличения затрат памяти [5].

Библиографический список

1. Domain-specific. – URL: http://en.wikipedia.org/wiki/Domain-specific_language

2. ANTLR. – URL: <http://www.antlr.org/>
3. Lex-yacc. – URL: <http://rus-linux.net/lib.php?name=/MyLDP/algol/lex-yacc-howto.html>
4. Scala. – URL: <http://www.scala-lang.org/node/12603>
5. Проблема синтаксического анализа текстов на внешних предметно-ориентированных языках. – URL: <https://research-journal.org/technical/problema-sintaksicheskogo-analiza-tekstov-na-vneshnix-predmetno-orientirovannyh-yazykax/>

Ю. Н. Косников, Т. Х. Хоанг

Пензенский государственный университет, Пенза, Россия

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ИЗОБРАЗИТЕЛЬНЫХ И ТОЧНОСТНЫХ СВОЙСТВ СМЕШИВАЮЩИХ ФУНКЦИЙ В ЗАДАЧАХ ВИЗУАЛИЗАЦИИ ПРОСТРАНСТВЕННЫХ ОБЪЕКТОВ

Представлены рекомендации по применению смешивающих функций для интерполяции и визуализации поверхностей, заданных множеством характерных точек (scattered data). Описан программный комплекс, позволяющий визуализировать результаты интерполяции и вычислять ее погрешность.

Форма объектов в системах визуализации часто задается скалярным полем, т.е. множеством неравномерно расставленных в пространстве характерных (опорных) точек. Графическая система компьютера поддерживает полигональную модель 3D-сцены. Для полигонального представления объекта по множеству опорных точек применяют интерполяционные методы. Анализ показывает, что предпочтение имеет интерполяция на основе смешивающих функций (СФ) [1]. Если каждому узлу интерполяции (каждой опорной точке) сопоставить некоторую СФ, то координаты текущей точки поверхности объекта определяются «вкладами» (коэффициентами влияния) СФ и зависят от удаления текущей точки от опорных точек. Тогда интерполяция в общем случае выглядит следующим образом:

- в явной форме в декартовой системе координат (СК) x, y, z

$$z = \sum_{i=1}^N \lambda_i \varphi(r_i),$$

- в параметрической системе координат u, v

$$x = \sum_{i=1}^N \lambda_{xi} \varphi(r_i), \quad y = \sum_{i=1}^N \lambda_{yi} \varphi(r_i), \quad z = \sum_{i=1}^N \lambda_{zi} \varphi(r_i),$$

где $\varphi(r_i)$ – смешивающая функция, зависящая от расстояния r_i между узлами интерполяции и текущей опорной точкой; $\lambda_i, \lambda_{xi}, \lambda_{yi}, \lambda_{zi}$ – коэффициенты влияния i -й СФ на текущую точку поверхности объекта; x_i, y_i – координаты проекции i -й опорной точки на плоскость аргументов; u_i, v_i – координаты i -й опорной точки в параметрической системе координат; N – количество опорных точек.

Расстояние r_i является той или иной функцией координат опорной точки в зависимости от выбранной системы координат. Например, в декартовой СК

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2},$$

в параметрической СК, параметрами которой являются шаги вдоль линий криволинейной координатной сетки uv , расположенной на поверхности объекта,

$$r_i = \sqrt{(u - u_i)^2 + (v - v_i)^2},$$

в параметрической СК, параметрами которой являются сферические координаты φ (азимутальный угол) и θ (полярный угол),

$$r_i = \arccos(\cos\theta \cdot \cos\theta_i + \sin\theta \cdot \sin\theta_i \cdot \cos(\varphi - \varphi_i)).$$

Выбор формы записи интерполянта и выбор СФ зависят от особенностей реконструируемой поверхности. Форма записи выбирается исходя из закона расположения опорных точек. Если они задают незамкнутую поверхность, рационально применить интерполянт в явной форме записи, если форма поверхности более сложная и характеризуется многозначностью или если поверхность замкнутая, следует применить параметрическую форму записи интерполянта. В литературе встречается неявная (общая) форма записи интерполянта [2]:

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

но ее нельзя признать удачной для визуализации объекта. Для нахождения промежуточных точек поверхности в этом случае нужно организовать перебор значений трех координат во всей области определения интерполянта и подстановку их в функцию-интерполянт. Если очередная точка обращает левую часть интерполянта в ноль (с заданной погрешностью), то эта точка принадлежит поверхности объекта. Такой переборный алгоритм визуализации весьма затратен. В связи с этим нужно стараться применять явную или параметрическую форму описания интерполянта. Рекомендации по переходу к таким формам представления интерполянта даны, например, в работах [3,4].

Относительно выбора СФ можно дать только общие рекомендации:

– чтобы уменьшить число слагаемых функции-интерполянта, нужно минимизировать области влияния опорных точек. Для этого следует применять СФ, локализованные в пространстве. Из общеупотребительных СФ к ним относятся полиномы Бернштейна (для сплайн-интерполяции) [5] и некоторые виды радиальных базисных функций (РБФ), например, инверсный квадратик, инверсный мультиквадрик, гауссиан, S^4 Матерн [6];

– если влияние опорной точки на текущую точку определяется кратчайшим расстоянием между ними, то при последовательном нахождении промежуточных точек поверхности это расстояние изменяется на каждом шаге вычислений. Для упрощения вычислений имеет смысл использовать СФ, влияние которых на текущую точку поверхности объекта находится раздельно по координатам-аргументам интерполянта. Тогда на каждом шаге вычислений будет изменяться только один компонент СФ. Это свойственно для СФ ортогонального базиса (СФОБ). В качестве таких СФ (*bf*) можно предложить

$$\text{параболическую вогнутую СФ: } bf = (1 - r_x)^2 \cdot (1 - r_y)^2,$$

$$\text{СФ-аналог гауссиана: } bf = (1 - r_x^2) \cdot (1 - r_y^2),$$

$$\text{биквадратную СФ: } bf = (r_x^2 - 1)^2 \cdot (r_y^2 - 1)^2.$$

Относятся к таким СФ и полиномы Бернштейна;

- для ускорения процесса интерполяции следует применять СФ в форме степенных полиномов. Такие СФ могут быть вычислены по «быстрым» конечно-разностным формулам. Форму полиномов имеют СФ ортогонального базиса, некоторые РБФ и полиномы Бернштейна [6,7].

Окончательный выбор СФ возможен на основе данных эксперимента, так как аналитическое исследование функции-интерполянта в общем случае невозможно. Причиной является большое число слагаемых интерполянта (сотни и тысячи) и сложный для анализа вид многих СФ, включающих радикалы, экспоненты, логарифмы и другие ресурсоемкие компоненты. На кафедре ИВС разработан программный комплекс, который служит для исследования и визуальной оценки изобразительных возможностей различных разновидностей СФ, изучения влияния параметров СФ на качество интерполяции, выбора количественных значений параметров СФ. Структура программного комплекса приведена на рис. 1.



Рис. 1. Структура комплекса программ для экспериментального исследования смешивающих функций

Средствами программного комплекса представляются для визуального анализа протяженные и локализованные в пространстве замкнутые и незамкнутые поверхности. Вид и параметры СФ, применяемых для интерполяционного моделирования поверхностей, задаются пользователем. Изображения формируются в каркасной и полнографической (с тенями и текстурами) форме. Пример изображения в главном окне программного комплекса показан на рис. 2.

Программы позволяют оценить погрешность интерполяции. В качестве оценки выбрано среднеквадратическое отклонение (СКО) полученной поверхности от эталонной, в качестве которой выбрана поверхность Франке [8]. Результаты оценки некоторых РБФ сведены в табл. 1. Значения СКО зависят не только от вида РБФ, но и от плотности расположения опорных точек. Для эксперимента использована область определения аргументов интерполанта, образующая квадрат единичной площади, поэтому количество N опорных точек в табл. 1 по сути является плотностью их расположения.

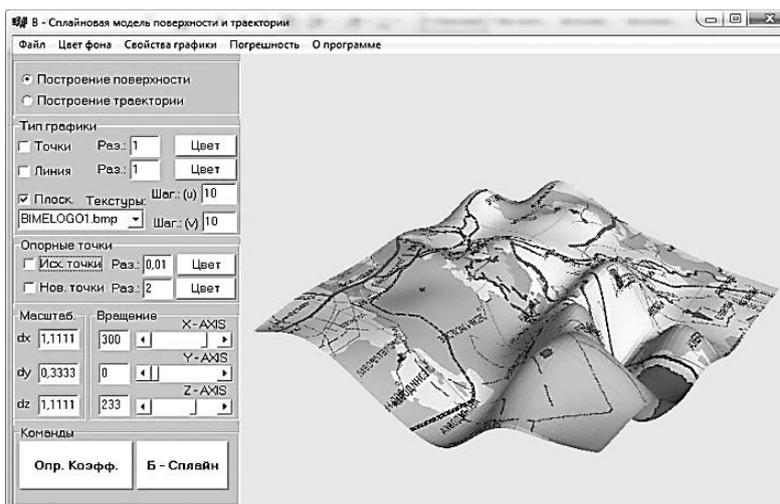


Рис. 2. Изображение протяженной поверхности с текстурой и тенями

Таблица 1

| Вид РБФ | Значения СКО при различных количествах опорных точек (N) | | | |
|------------------------------------|--|---------|---------|---------|
| | $N=40$ | $N=60$ | $N=80$ | $N=100$ |
| Гауссиан | 0,02346 | 0,01284 | 0,01245 | 0,00604 |
| Инверсный мультиквадрик | 0,02624 | 0,01542 | 0,00717 | 0,00257 |
| Мультиквадрик | 0,01902 | 0,01209 | 0,00601 | 0,00217 |
| Инверсный квадратик | 0,02989 | 0,01895 | 0,00971 | 0,00262 |
| Полигармонические сплайны | | | | |
| Кубик | 0,02132 | 0,01735 | 0,00805 | 0,00457 |
| Плоский сплайн (Thin Plate Spline) | 0,02591 | 0,01565 | 0,00681 | 0,00680 |
| C^0 Матерн | 0,04889 | 0,04455 | 0,03752 | 0,01871 |
| C^2 Матерн | 0,02407 | 0,01789 | 0,00996 | 0,00425 |
| C^4 Матерн | 0,02192 | 0,01384 | 0,00638 | 0,00411 |

Результаты оценки точности интерполяции для предложенных СФОБ приведены в табл. 2.

Таблица 2

| Наименование СФОБ | Значение СКО |
|----------------------------|------------------------|
| Параболическая вогнутая СФ | 0,006682 |
| СФ-аналог гауссиана | $2,7458 \cdot 10^{-5}$ |
| Биквадратная СФ | $5,646 \cdot 10^{-5}$ |

Приведенные в таблицах данные показывают, что с помощью программного комплекса можно выбирать вид и значения параметров СФ. Программы комплекса защищены свидетельствами о государственной регистрации [9, 10].

Библиографический список

1. Александрова, Н. В. Смешивающие функции в геометрическом моделировании и визуализации поверхностей свободных форм / Н. В. Александрова, А. П. Зимин, Ю. Н. Косников, Т. Х. Хоанг // XXI век: итоги прошлого и проблемы настоящего плюс. Сер.: Технические науки. Информационные технологии. – Пенза, 2015. – № 03 (25). – Т. 1. – С. 51–60.
2. Reconstruction and Representation of 3D Objects with Radial Basis Functions / J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, T. R. Evans // Proceedings of the 28th annual conference on Computer graphics and interactive techniques. – New York, 2001. – P. 67–76.
3. Косников, Ю. Н. Методика и технология компьютерного моделирования поверхностей свободных форм с применением радиальных базисных функций / Ю. Н. Косников // XXI век: итоги прошлого и проблемы настоящего плюс. Сер.: Технические науки. Информационные технологии. – Пенза, 2014. – № 03(19). – С. 176–183.
4. Kosnikov, Yu. N. Morphing of spatial objects in real time with interpolation by functions of radial and orthogonal basis / Yu. N. Kosnikov, A. V. Kuzmin, T. H. Hoang // Journal of Physics: Conference Series. – IOP Publishing. – 2018. – Series 1015. – № 032066. – URL: <http://iopscience.iop.org/article/10.1088/1742-6596/1015/3/032066/meta>
5. Роджерс, Д. Математические основы машинной графики / Д. Роджерс, Дж. Адамс. – М.: Мир, 2001. – 604 с.
6. Косников, Ю. Н. Моделирование и визуализация неаналитических поверхностей / Ю. Н. Косников, Т. Х. Хоанг // XXI век: итоги прошлого и проблемы настоящего плюс. Сер.: Технические науки. Информатика, вычислительная техника и управление. – Пенза, 2016. – № 06 (34). – С. 129–136.
7. Skala, V. A Practical Use of Radial Basis Functions Interpolation and Approximation / V. Skala // Revista de Investigacion Operacional. – 2016. – Vol. 37(2). – P. 137–145. – URL: <http://rev-inv-ope.univ-paris1.fr/volumes-since-2000/volume-37-2016>
8. Franke, R. Scattered Data Interpolation: Test of some method / R. Franke // Mathematics of Communication. – 1982. – Vol. 38. – P. 181–200.
9. Свидетельство № 2017613959. Программа визуализации протяженных геометрических форм, заданных скалярным полем: Свидетельство о государственной регистрации программы для ЭВМ / Ю. Н. Косников (RU), Т. Х. Хоанг (VN). – Оpubл. 04.04.2017.

10. Свидетельство № 2018613102. Программа моделирования и визуализации протяженных геометрических форм с применением радиальных базисных функций: Свидетельство о государственной регистрации программы для ЭВМ / Ю. Н. Косников (RU), Т. Х. Хоанг (VN). – Оpubл. 02.03.2018.

К. Б. Новиков, А. Ю. Боровиков

ПФ ФГУП «НТЦ «Атлас», Пенза, Россия

РАССМОТРЕНИЕ ВОЗМОЖНОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ МОДУЛЯ ДОВЕРЕННОЙ ЗАГРУЗКИ ОПЕРАЦИОННОЙ СИСТЕМЫ

Рассмотрено понятие доверенной загрузки операционной системы, предложен современный подход к программной реализации модуля доверенной загрузки.

Одной из важнейших задач информационной безопасности является пресечение несанкционированного доступа (НСД) к конфиденциальной информации. На персональных компьютерах для обеспечения невозможности совершения НСД применяются различные способы защиты, такие как обязательная аутентификация при загрузке в ОС, шифрование данных, разграничение доступа к охраняемым данным посредством системы разграничения прав пользователей. Перечисленные способы обеспечивают защиту информации после загрузки операционной системы.

При этом загрузка операционной системы является местом уязвимости и возможной целью атакующего при попытке совершить НСД, так как на данном этапе вышеуказанные способы защиты еще не вступили в силу. Наиболее простым способом атаки является загрузка ОС с внешнего носителя, что позволит злоумышленнику получить доступ к защищенной информации. В качестве защиты от подобной атаки часто применяется опломбирование системного блока и запрет на работу с неучтенными носителями информации в рабочем месте, что не является достаточным для гарантированного обеспечения защиты от НСД.

На настоящий момент основным способом защиты от загрузки ОС с неучтенного носителя является обеспечение «доверенной загрузки».

Доверенная загрузка – загрузка операционной системы, производимая при успешной аутентификации оператора ЭВМ и исключительно с учтенного носителя. Средства, обеспечивающие доверенную загрузку (модули доверенной загрузки), должны иметь соответствующие разрешительные документы для применения на территории Российской Федерации. Модули доверенной загрузки могут быть реализованы на аппаратно-программном или программном уровне.

Аппаратно-программная реализация требует использования внешнего устройства, подключаемого к контролируемой ЭВМ по шине PCI/PCIe. При установке аппаратно-программные модули доверенной загрузки (АПМДЗ) могут потребовать инсталляции специального ПО на учтенный носитель информации или записи дополнительного модуля в образ BIOS, используемый в ЭВМ. Устройству передает управление базовая система ввода-вывода после инициализации аппаратных устройств в составе ЭВМ. Доверенный загрузчик выполняет определенный разработчиком набор функций, исходя из результата выполнения которых будет принято решение, выполнять загрузку операционной системы или нет.

Недостатком аппаратно-программной реализации является сложность ее применения во встраиваемых устройствах, к габаритам которых предъявляются жесткие требования. Наличие платы АПМДЗ в ЭВМ и, в случае необходимости, переходника для платы АПМДЗ может значительно увеличить размеры устройства (его корпуса).

Программная реализация требует записи бинарного модуля или полного замещения образа BIOS в составе материнской платы. Программный модуль доверенной загрузки

(ПМДЗ) получает управление при выполнении BIOS в составе материнской платы и не требует обязательного наличия дополнительных аппаратных устройств для своего функционирования. Недостатком программной реализации является необходимость внедрения или разработки собственного образа BIOS, который будет размещен на материнской плате.

Внедрение модуля в бинарный образ BIOS является трудоемкой задачей, кроме того данный вариант реализации не гарантирует отсутствие ошибок или уязвимостей за пределами внедренного модуля по причине невозможности проведения их анализа.

Рассмотрим возможность разработки ПМДЗ с целью полного замещения образа BIOS. Для этого необходимо иметь исходный программный код, из которого будет собран BIOS, для возможности его расширения функциями доверенной загрузки. При этом ПМДЗ должен реализовывать базовые функции BIOS (инициализация аппаратных компонентов системы, поддержка прерываний BIOS, передача управления загрузчику операционной системы на доверенном носителе). Общий алгоритм работы описанного ПМДЗ представлен ниже (рис. 1).

С учетом вышеуказанных требований, наиболее подходящей основой для разработки ПМДЗ является ПО проекта «Coreboot».

«Coreboot» – проект с открытым исходным кодом, предназначенный для сборки BIOS различных аппаратных платформ [1]. На текущий момент существуют готовые реализации и наборы настроек для систем на базе x86-архитектуры, ARM, MIPS, SPARC. Поддерживаются материнские платы с центральными процессорами фирм Intel и AMD, реализуется включение в состав BIOS и исполнение бинарных вставок от производителей системной логики ЦП для инициализации аппаратных компонентов ЭВМ.

Доступ к исходному коду проекта «Coreboot» позволяет как проанализировать его на предмет уязвимостей, так и включить в состав функции доверенной загрузки – аутентификация пользователей (в том числе с использованием внешних считывателей ключей), контроль целостности BIOS и носителей информации в составе системы, а также их учет. Наличие исходного кода позволяет провести сертификационные работы по требованиям ФСТЭК для получения разрешительных документов и применения ПМДЗ на территории РФ.

Данные, изменяемые в процессе работы BIOS (база логинов, паролей, прав пользователей), могут быть сохранены во внутренней CMOS-памяти или на внешнем носителе информации [2].

Непосредственно передача управления загрузчику ОС должна быть осуществлена с помощью включения в состав «Coreboot» дополнительной программы – «полезной нагрузки». На данный момент существует ряд «полезных нагрузок», совместимых с «Coreboot» и предоставляющих доступ к исходному программному коду: «SeaBIOS», «FILO», «tint». В зависимости от требуемых функций, имеется возможность как выполнения загрузки ОС, так и реализация на базе «полезной нагрузки» собственного ПО и его выполнение. При этом разработчик ограничивается исключительно размерами носителя, содержащего образ BIOS.

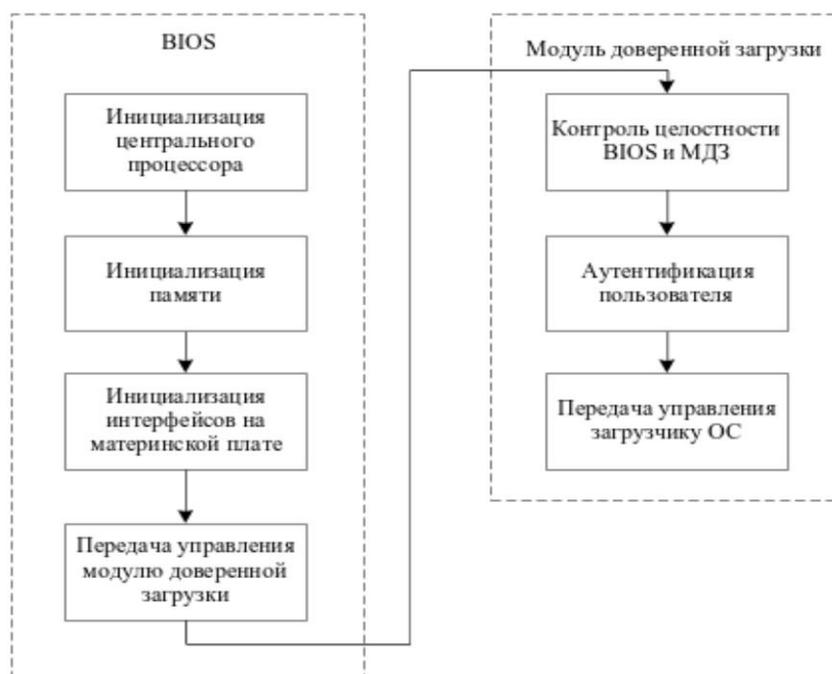


Рис. . Алгоритм работы программного модуля доверенной загрузки

Программная реализация модуля доверенной загрузки ОС может быть выполнена с использованием других существующих средств. Аналогом проекта «Coreboot» является проект «Tianocore» на базе системы сборки «EDK II». В отличие от «Coreboot», результатом сборки которого является Legacy BIOS, результатом сборки «Tianocore» является образ UEFI BIOS. Проект «Tianocore» разрабатывается фирмой Intel и поддерживает большое число аппаратных платформ данной фирмы. Недостатком проекта является высокий процент бинарного кода, из которого производится сборка образа BIOS. ФСТЭК не имеет нормативных документов, содержащих требования к сертификации программного обеспечения в бинарном коде, что не позволяет провести сертификационные работы по полученному образу BIOS. Кроме того, программную логику, реализованную в бинарном коде, сложно модифицировать, в том числе для добавления функций доверенной загрузки операционной системы.

Таким образом, в настоящее время является возможным разработать BIOS, реализующий функции доверенной загрузки операционной системы. Реализация программного модуля доверенной загрузки на основе ПО проекта «Coreboot» является наиболее современным подходом, позволяющим в дальнейшем провести сертификационные работы относительно полученного образа BIOS по требованиям ФСТЭК. Успешное проведение данных работ позволит применять модуль доверенной загрузки на территории Российской Федерации.

Библиографический список

1. Zimmer, V. Embedded Firmware Solutions / V. Zimmer // A-Press One. – 2015. – P. 55.
2. Sallihun, D. BIOS Ninjutsu Uncovered / D. Sallihun // A-Press One. – 2006. – P. 720.

Е. Н. Прошкина

Пензенский государственный университет, Пенза, Россия

ОНТОЛОГИЧЕСКИЙ ИНЖИНИРИНГ В ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ

Проектирование информационной системы предприятия всегда начинается с исследования информационных процессов и структур. В статье предлагается выполнить исследование с помощью фреймворка Захмана, который позволяет получить онтологию проектируемой информационной системы. Результаты исследования представляются в виде матрицы. Каждая колонка матрицы представляет собой отдельный аспект информационной системы, представленный с разной степенью детализации и с точки зрения разных заинтересованных лиц.

Введение

В настоящее время трудно себе представить функционирование любого современного предприятия без поддержки информационной системы, разработка которой всегда основывается на исследовании информационных процессов и структур предприятия. Фреймворк Захмана – это один из самых старых архитектурных фреймворков, который является отраслевым стандартом. В основе данного фреймворка лежит таксономия артефактов таких, как данные, функциональность, модели, спецификации, документы [1]. Фактически он представляет собой онтологию верхнего уровня, которую разработчик может расширять и уточнять, получая в результате онтологию, описывающую конкретную систему.

Постановка задачи

Задачей нашего исследования является исследование информационных процессов и структур предприятия с использованием комбинации системного структурного и объектно-ориентированного подходов и представление результатов в виде единой онтологии с помощью фреймворка Захмана.

Жизненный цикл создания онтологии

Жизненный цикл (ЖЦ) любой онтологии состоит из ряда процессов, в ходе которых онтология зарождается, специфицируется, адаптируется, развертывается, используется и поддерживается. Происходят эти процессы параллельно или последовательно, однажды или повторяются несколько раз, частично зависит от того, как выполнялось создание онтологии. Поскольку онтологии создаются для разных задач, отдельные стадии жизненного цикла могут отсутствовать для одних онтологий и присутствовать для других. Данный факт не позволяет создать единую общую модель ЖЦ онтологии с четко обозначенной последовательностью этапов ЖЦ. ЖЦ конкретных онтологий являются упрощенными частными случаями, зависящими от специфики создания конкретной онтологии. Оценка онтологий производится на всех стадиях ЖЦ, что позволяет установить, насколько онтология удовлетворяет требованиям последующей стадии. Рассмотрение упрощенного ЖЦ позволяет выделять стадии, общие для всех онтологий:

- стадия формирования требований,
- стадия онтологического анализа,
- стадия проектирования онтологии,
- стадия проектирования информационной системы, основанной на онтологии,
- стадия разработки онтологии,
- стадия разработки информационной системы, основанной на онтологии,
- стадия развертывания,
- стадия промышленной эксплуатации.

Как правило, онтология в процессе своего ЖЦ проходит через выделенные стадии более одного раза. Идентификация стадий ЖЦ онтологии позволяет кластеризовать действия вокруг целей, входов и выходов узнаваемого типа. Более того, модель ЖЦ наглядно демонстрирует зависимость одних стадий ЖЦ от других, например, качество онтологии

Аппаратно-программное обеспечение информационно-вычислительных систем и систем управления напрямую зависит от того, насколько грамотно были сформулированы требования к ней. Зависимости между стадиями инвариантны для всех онтологий, несмотря на их различия между собой.

Как показывает анализ стадий ЖЦ онтологий, все они хорошо коррелируют с теми стадиями, которые выделены при проектировании информационных систем. И это не случайно, поскольку любое проектирование это, по существу, создание онтологии.

Как и в случае проектирования информационных систем, здесь используется концепция быстрого прототипирования, а специфика проявляется в конкретных процессах. При этом:

- планирование выполняется до начала собственно разработки;
- контроль и гарантии качества осуществляются в процессе разработки;
- большая часть операций по накоплению знаний и их оценке выполняется на стадии концептуализации для того, чтобы предотвратить распространение ошибок на фазу реализации;
- интеграция не должна рассматриваться как интеграция на стадии реализации, напротив, она выполняется в процессе разработки.

Представление информационных процессов и структур предприятия с помощью фреймворка Захмана

Рассмотрим проектирование информационной системы с помощью фреймворка Захмана. Согласно фреймворку Захмана функционирование организации можно описать в терминах ответа на шесть вопросов [2]:

- используемые данные (что?);
- процессы и функции (как?);
- места выполнения процессов (где?);
- организации и персоналии (кто?);
- управляющие события (когда?);
- цели и ограничения, определяющие работу системы (почему?).

Ответы на эти вопросы можно давать с разной степенью детализации: уровень контекста, уровень бизнес-описаний, системный уровень, технологический уровень, технический уровень, уровень реальной системы. В результате формируется матрица размером 6 × 6, в каждой клетке которой ставятся в соответствие модели и артефакты (табл. 1).

Таблица 1

| | Что? | Как? | Где? | Кто? | Когда? | Почему? | |
|-------------------------------|------------------------------|-------------------------------------|------------------------------------|---------------------------|-----------------------------|---------------------------|----------------------|
| Контекст | Список основных сущностей | Основные бизнес-процессы | Размещение организации | Внешние организации | Список событий | Бизнес-стратегия | Аналитики |
| Бизнес-модель | Отношения между сущностями | Подробное описание бизнес-процессов | Система логистики | Модель потоков работ | Базовый график работ | Дерево целей | Топ-менеджеры |
| Системная модель | Концептуальные модели данных | Архитектура приложений | Архитектура распределенной системы | Интерфейсы пользователя | Модель работы с событиями | Бизнес-правила | Архитекторы |
| Технологическая модель | Физическая модель данных | Программно-аппаратная архитектура | Технологическая архитектура | Архитектура представления | Алгоритмы обработки событий | Правила обработки событий | Разработчики |

Окончание табл. 1

| | Что? | Как? | Где? | Кто? | Когда? | Почему? | |
|--|-------------|-------------|-------------|-------------|---------------|----------------|--|
|--|-------------|-------------|-------------|-------------|---------------|----------------|--|

| Детальное описание | Спецификации форматов данных | Исполняемый код | Архитектура сети | Роли и права пользователей | Обработка событий прерываниями | Алгоритмы работы системы | Администраторы |
|-----------------------------|------------------------------|------------------------------|------------------------|---------------------------------------|----------------------------------|--------------------------|----------------|
| Функционирующая организация | Данные | Реализуемая функциональность | Сетевая инфраструктура | Организационная структура организации | История функционирования системы | Реализуемые стратегии | Пользователи |

Выполним исследование информационных процессов и структур предприятия с использованием системного структурного и объектно-ориентированного подхода и представим результаты исследования с помощью фреймворка Захмана. Первая колонка определяет используемые в системе данные. На верхнем уровне необходимо выделить основные сущности, построив концептуальную модель. На втором уровне будем использовать данные сущности для построения логической модели данных в нотации IDEF1X. На третьем уровне полученную модель приведем к нормализованной форме, определим все атрибуты и ключи. Четвертый уровень представляет собой физическую модель данных в нотации IDEF1X. Пятый уровень включает описание модели на языке управления данными для формирования таблиц. На последнем уровне описываются фактические наборы данных.

Колонка «процессы и функции» содержит описания последовательностей операций. Так первому уровню соответствует простое перечисление ключевых бизнес-процессов организации, построение контекстной диаграммы в нотации IDEF0. Второй уровень содержит детальные описания бизнес-процессов в нотациях IDEF0, DFD, IDEF3. На третьем уровне разрабатываются архитектуры приложений в виде диаграмм классов UML. На четвертом уровне прорабатывается программно-аппаратная архитектура в виде диаграммы развертывания и диаграммы компонентов языка UML. На пятом уровне представляется программный код и на шестом – исполняемые модули.

Колонка «места выполнения процессов» определяет пространственное распределение компонент системы и сетевую инфраструктуру [3]. На контекстном уровне оказывается достаточным определить расположение всех основных объектов организации (организационная структура организации). На следующем уровне эти объекты объединяются в модель со связями, характеризующими взаимодействие между ними, т.е. строится система логистики организации. На третьем уровне осуществляется привязка компонент информационной системы к узлам сети с помощью диаграммы развертывания языка UML. Четвертый уровень служит для определения физической реализации в терминах аппаратных платформ, системного программного обеспечения и промежуточного ПО. На пятом уровне определяются используемые протоколы и спецификации каналов связи. Шестой уровень описывает функционирование реализованной сети.

Колонка «организации и персоналии» определяет участников процесса. На первом уровне определяется список организаций-партнеров, подразделений организации и выполняемые ими функции. Это может быть выполнено в текстовой форме. На следующем уровне составляется модель потоков работ, выполненная в виде диаграммы прецедентов языка UML. Далее последовательно определяются интерфейсы пользователя для каждой роли пользователя, выделенной в системе. На четвертом уровне разрабатывается архитектура представления, описывающая правила доступа к отдельным объектам. На пятом уровне детально прорабатывается политика безопасности системы. На шестом уровне строится подробная организационная структура организации.

Колонка «управляющие события» определяет временные параметры бизнес-процессов и работы системы. На контекстном уровне составляется список значимых для функционирования системы событий. Затем на уровне бизнес-модели разрабатывается базовый график работ. Он может быть представлен, например, в виде диаграмм функцио-

Аппаратно-программное обеспечение информационно-вычислительных систем и систем управления нально-стоимостного анализа в нотации IDEF0. На третьем уровне разрабатывается модель работы с выделенными на предыдущих уровнях событиями. Для технологического уровня алгоритмы обработки событий могут быть представлены в виде диаграмм деятельности языка UML. Пятый уровень определяет программную реализацию процесса обработки событий. На шестом уровне история функционирования системы может быть представлена в форме записей в лог-файлах.

Колонка «цели и ограничения» описывает мотивации и задает порядок перехода от задач бизнеса к требованиям, предъявляемым к отдельным элементам системы. Исходной точкой является бизнес-стратегия, которая последовательно транслируется в дерево целей и бизнес-план. На уровне системной модели разрабатываются бизнес-правила и ограничения для реализации бизнес-процессов. На следующем уровне задаются правила обработки событий, имеющих место в системе. На уровне детального описания с помощью диаграмм деятельности языка UML представляются алгоритмы работы системы, которые затем трансформируются в реализуемые стратегии.

В последней колонке таблицы приведены заинтересованные лица, с точки зрения которых и рассматриваются обозначенные шесть вопросов.

Последовательно заполняя ячейки таблицы, можно выполнить исследование информационных процессов и структур организации, а также описать поведение проектируемой информационной системы. Однако рассматриваемый фреймворк в явном виде не позволяет описывать поведение системы в динамике, поскольку каждый элемент таблицы не может содержать как описание существующего состояния, так и целевого.

Данный фреймворк не накладывает ограничений на использование тех или иных аппаратных и программных платформ и инструментальных средств разработки. В данной работе предлагалось использовать в частности стандарты IDEF0, DFD, IDEF3, IDEF1X, которые поддерживаются различными инструментальными средствами, в том числе и свободно распространяемыми и язык моделирования информационных процессов UML.

Заключение

При разработке информационных систем, как правило, применяется синтетическая методика, т.е. деятельность предприятия описывается с помощью системного структурного подхода с использованием стандартов IDEF0, DFD, IDEF3, а проектирование программного обеспечения ведется с помощью объектно-ориентированного подхода на языке UML. Эти два описания обычно трудно собрать воедино. Использование фреймворка Захмана при исследовании информационных процессов и структур организации позволяет получить целостное представление о её деятельности и о разрабатываемой информационной системе, создать онтологию разрабатываемой системы, а также избежать противоречий в описании.

Библиографический список

1. Zachman, J. A. John Zachman's Concise Definition of the The Zachman Framework / J. A. Zachman // Zachman International. 2008. – URL: <http://www.zachmaninternational.com/concise%20definition.pdf>
2. Данилин, А. Архитектура и стратегия / А. Данилин, А. Слюсаренко. – М. : Интернет-университет информационных технологий, 2005. – 504 с.
3. Сафронов, А. А. Сравнительный анализ методологий построения архитектуры предприятий / А. А. Сафронов, Л. З. Давлеткиреева, В. Н. Макашова // Современная техника и технологии. – 2014. – № 1. – URL: <http://technology.snauka.ru/2014/01/2721> (дата обращения: 01.10.2017).

Д. А. Родионов

АНАЛИЗ ВОЗМОЖНОСТЕЙ МИКРОСХЕМ ФИРМЫ FTDI

Выполнен анализ возможностей микросхем фирмы FTDI и выбор наиболее эффективного варианта микросхемы для использования ее в качестве базовой при разработке аппаратно-программного комплекса для тестирования РЭА.

Сегодня уже никого не удивит наличием USB-интерфейса у персональных компьютеров (ноутбуков). Во многих случаях он становится практически единственным проводным коммуникационным интерфейсом, позволяющим подключать к компьютеру различные внешние устройства – цифровые видеокамеры и фотоаппараты, принтеры и сканеры, внешние устройства хранения данных. Простая в использовании, шина USB практически вытеснила своих предшественников – параллельную шину (LPT) и последовательный интерфейс (COM – порт).

Большинство современных персональных компьютеров обязательно оснащается интерфейсом USB и может уже не иметь COM или LPT порта. Такая ситуация отрицательно сказывается на производителях приборов и систем, разработанных ранее и не имеющих возможности подключения к современным компьютерам по USB. То же самое можно сказать и о конечных пользователях, покупающих, например, ноутбук и не имеющих возможности подключить к нему проверенную временем аппаратуру, которая не оснащена интерфейсом USB.

Очевидный выход из данной ситуации – оснащение выпускаемой техники новым интерфейсом. В такой ситуации присутствуют отрицательные стороны как для производителя, так и для конечного потребителя, т.е. для нас с Вами. И если конечный потребитель может использовать специально выпускаемые переходные кабели USB-RS232 или USB-LPT, то производителю техники придется затратить куда больше усилий для разработки нового изделия. Причем изменения коснутся не только аппаратной части изделия, но и программной, как верхнего (хост), так и нижнего (микроконтроллер) уровня.

Что же делать, если в случае серийно выпускаемого устройства возникает потребность в короткие сроки оснастить его интерфейсом USB? Как, не вникая в подробности протокола USB, не затрачивая уйму времени на изучение драйверов различных классов и пытаясь приспособить их под собственные задачи, реализовать такой простой для конечного потребителя интерфейс?

В таких случаях самым очевидным выходом является использование аппаратных мостов USB.

Одной из ведущих компаний по разработке и производству таких специализированных микросхем является фирма FTDI. Компания была основана в 1992 году в Шотландии. Основной сферой деятельности являлась разработка чипсетов для материнских плат персональных компьютеров IBM. В 1996 году, после принятия первой спецификации USB 1.0, основной сферой деятельности FTDI стала разработка аппаратных мостов USB и драйверов для них. Практика компании показала, что наиболее популярными были и остаются аппаратные мосты с интерфейсом UART и параллельным интерфейсом FIFO. Именно аппаратные мосты "USB-UART" и "USB-FIFO" являются основной продукцией компании. Тем не менее, кроме них также выпускаются концентраторы USB, готовятся к производству хост-контроллеры, разрабатываются микросхемы нового аппаратного моста с поддержкой высокоскоростного режима передачи.

Серийное производство аппаратных мостов компания FTDI начала с выпуска микросхем FT8U232 "USB-UART" и FT8U245 "USB-FIFO", которые оказались достаточно удачным решением, востребованным на зарождающемся рынке микросхем с интерфейсом USB. Эти микросхемы имели только один режим работы, т.е. непосредственное преобразование "USB-UART" и "USB-FIFO", большое число внешних компонентов, включая кварцевый

резонатор и энергонезависимую память для хранения идентификаторов USB и настроек режима работы.

Возможности следующего поколения микросхем FT232B "USB–UART" и FT245B "USB–FIFO" были расширены путем добавления нового режима работы Bit Bang. В данном режиме выходной интерфейс микросхем реализует 8 независимых линий ввода/вывода. В таком режиме мост может, например, управлять работой реле без дополнительного микроконтроллера. Кроме того, была изменена схемотехника кристалла. В него был интегрирован контур "power-on-reset". В остальном никаких дополнительных доработок не было.

2006 год был ознаменован появлением нового поколения микросхем – FT232R и FT245R, в которых были воплощены пожелания многих разработчиков: интегрированные на кристалл энергонезависимая память для хранения настроек режима работы и идентификаторов USB, встроенный тактовый генератор и некоторые пассивные компоненты.

Встроенная EEPROM с выделенной областью для записи пользовательской информации и уникальный идентификатор в сумме дают возможность создавать на базе новых микросхем FTDI аппаратные ключи для защиты от несанкционированного доступа. Для работы с новой функцией была разработана новая библиотека, которая работает совместно с драйвером D2XX.

По сравнению с предыдущими версиями претерпел кардинальные изменения встроенный преобразователь напряжения 3.3 В. В новых версиях микросхем его нагрузочная способность достигла 50 мА, что в 10 раз больше, чем у предыдущих версий. Внесены усовершенствования и в режим Bit Bang. Теперь изменение состояния линий ввода/вывода может сопровождаться контрольными сигналами чтения или записи. Новое поколение микросхем рассчитано на работу в промышленном температурном диапазоне от –40 °С до +85 °С. Новые микросхемы выпускаются в двух типах корпусов - SSOP 28 и QFN 32. Оба типа корпуса отвечают требованиям стандарта RoHS.

Микросхемы серии FT232R имеют дополнительные возможности по сравнению с микросхемами FT245R.

Изменилось функциональное назначение выводов TXLED, RXLED, TXDEN, PWREN, SLEEP. В новой модификации эти выводы носят название CBUS. Настройка выводов CBUS хранится в EEPROM и выбирается индивидуально для каждого из выводов. По умолчанию их функциональное назначение соответствует перечисленным выше сигналам. Другими возможными конфигурациями шины CBUS могут быть режим Bit Bang или возможность транслирования тактового сигнала от встроенного генератора.

В режиме Bit Bang могут работать 4 из 5 линий CBUS. Они могут служить дополнительными линиями ввода/вывода. На скорость обмена по линиям CBUS будет влиять загрузка основного канала USB–UART.

Для передачи тактовых сигналов от встроенного генератора на внешние устройства могут быть задействованы все 5 выводов шины CBUS. Частота выходного сигнала на любом из выводов может быть задана равной 6, 12, 24 или 48 МГц независимо от конфигурации других выводов. Такая возможность позволяет отказаться от использования в схеме дополнительных элементов, таких как, например, тактовые генераторы и распределители тактовых сигналов.

В ряду продукции фирмы FTDI также стоит отметить микросхему FT2232C. На ее базе можно реализовать два независимых канала "USB–UART" или "USB–FIFO". Выходные интерфейсы выбираются независимо друг от друга. Кроме основных функций, микросхемы данной серии могут быть использованы для загрузки конфигурации в микросхемы FPGA производства компании Altera или для эмуляции последовательных интерфейсов IIC, SPI или JTAG.

Итак, использование специализированных микросхем FTDI решает аппаратную проблему реализации интерфейса USB в целевом приборе. Наиболее эффективным вариантом является выбор микросхемы FT2232C для использования в качестве базы при разработке аппаратно-программного комплекса для тестирования РЭА. Основанием ее выбора послужило наличие функции для загрузки конфигурации в микросхемы FPGA, а также эмуляция интерфейса JTAG, что является неотъемлемой частью разрабатываемого комплекса.

1. Долгушин, С. Высокоскоростные контроллеры USB производства компании Cypress / С. Долгушин // Компоненты и технологии. – 2006. – № 6. – URL: <http://cyberleninka.ru>
2. Software Application Development D2XX Programmer's Guide, 2012. – URL: <http://ftdichip.com>
3. Application Note AN-130 FT2232C Used In An FT245 Style Synchronous FIFO Mode. – URL: <http://ftdichip.com>

Д. С. Смирнов

Пензенский государственный университет, Пенза, Россия

РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ВЕБ-САЙТОВ НА ОСНОВЕ ВЕРОЯТНОСТНОГО АВТОМАТА

Рассмотрены ошибки, возникающие во время работы веб-сайтов. Предложена реализация инструмента автоматизированного тестирования веб-сайтов на основе вероятностного автомата. Рассмотрены вопросы описания тестов, внутренние механизмы инфраструктуры тестирования, процесса тестирования и оценки результатов тестирования.

Веб-сайт является многопользовательской системой, где одновременно работают несколько пользователей. Во время работы могут возникать ситуации, когда взаимодействия одного пользователя с системой оказывают не запланированное влияние на взаимодействия другого пользователя с системой. Примером такой ошибки может быть ситуация с «потерянным обновлением», когда изменения одного пользователя перекрываются изменениями другого пользователя. Такие ошибки встречаются весьма часто. Они сильно зависят от конкретной бизнес логики приложения и их сложно классифицировать. Их объединяет, то что на этапе тестирования их очень сложно выявить, так как во время теста системой пользуется один человек.

Для выявления таких проблем проводят альфа и бета тестирования, но оба способа требуют привлечения значительного числа людей (Альфа- и бета-тестирование). Из автоматизированных средств, наиболее подходящим является проведение нагрузочного тестирования (J.D. Meier, 2007). Нагрузочное тестирование не может являться инструментом для решения поставленной проблемы, так как обладает рядом недостатков. Нагрузочное тестирование проводится в первую очередь для определения производительности системы, а не выявления ошибок. Те пользователи, что создаются во время тестирования в большинстве своем следуют одному жестко прописанному сценарию. Большинство инструментов для нагрузочного тестирования взаимодействуют с веб-сервером напрямую, а не взаимодействуя через веб-браузер. Все это снижает вероятность определения таких ошибок.

Для решения данных проблем, было разработано решение, которое лишено недостатков нагрузочного тестирования. Целью многомерного тестирования является выявления ошибок в условиях максимально приближенных к реальным. Многомерное тестирование так же, как и нагрузочное само создает необходимое число пользователей, поэтому не нужно привлекать дополнительных пользователей. Многомерное тестирование позволяет описать поведение пользователя максимально приближенного к поведению реального пользователя. Поведение пользователя описывается с помощью автомата (Lee. D, 1996). Взаимодействие с веб-браузером осуществляется с помощью специальных веб-драйверов на базе технологии Selenium, за счет взаимодействия с системой, которое будет аналогично тому, как взаимодействует конечный пользователь (Dima, 2014).

Процесс тестирования осуществляется следующим образом создаются виртуальные пользователи, для каждого определяется его персональный автомат. Автомат отвечает за поведение пользователя. При срабатывании перехода из одного состояния в другое состояние, пользователь совершает ряд действий, которые должны перевести тестируемую систему в новое состояние. После чего запускается проверка того, что состояние системы соответствует текущему состоянию автомата. Иными словами происходит тестирование в рамках, которого осуществляется сравнение ожидаемого поведения с действительным (Meszaros, 2007). Выбор совершаемый реальными пользователями носит случайный характер, поэтому для описания поведения пользователей используется вероятностный автомат (Финаев В.И., 2007).

Для описания вероятностного автомата потребуется задать состояния автомата, определить начальные состояние, определить алфавит, определить функции перехода и стохастическую матрицу вероятностей переходов (Миронов, Основные понятия теории вероятностных автоматов, 2015) (Миронов, Основные понятия теории вероятностных автоматов(часть 2), 2016).

Наиболее интересным компонентом автомата является функции перехода. Функции перехода не только переводят автомат в нужное состояние, но и совершают набор действий по переводу тестируемой системы в нужное состояние, а также совершают проверку соответствия между состоянием автомата и состоянием системы. Описание действий и проверок осуществляется с помощью программного кода. Каждая функция это отдельный функциональный тест разработанный в соответствии с такой технологией тестирования как NUnit (Andy Hunt, 2007).

Для определения количества пользователей в конкретный момент времени используются Профили нагрузки. У каждого Профиля пользователя, есть свой Профиль нагрузки. При этом разные пользовательские группы, могут использовать один и тот же Профиль нагрузки. Профиль нагрузки реализует некоторую функцию $y=f(t)$, где y это число пользователей в момент времени t . Профили нагрузки в том или ином виде реализованы в нагрузочном тестировании, наиболее популярными функциями – это постоянная функция, линейная функция и кусочно-постоянная функция (Erinle, 2015).

Для тестирования важно правильно настроить окружение для этого существует специальная фаза – это фаза инициализации. Во время фазы инициализации можно провести ряд мер, чтобы привести систему в нужное состояние. После завершения тестирования может потребоваться возможность удалить временные файлы, почистить базу данных и т.д. Для реализации данной возможности есть фаза очистки. Если для тестирования не нужны одна из этих фаз или обе сразу, то их можно и не определять, фазы являются опциональными. Реализация фаз осуществляется с помощью программного кода.

Разобрав как описываются тесты, перейдем к тому, какие внутренние механизмы работают во время тестирования. Основным таким механизмом являются Генератор пользователей. Генератор пользователей отвечает за регулирование количества пользователей во время тестирования. Это подразумевает создание и удаление пользователей, когда это необходимо. Каждый генератор работает с определенным Профилем пользователя, в соответствии с которым создаются пользователи и Профиль нагрузки, в соответствии с которым регулируется число этих пользователей.

Работа генератора основана на внутреннем таймере. Таймер с определенной периодичностью, формирует определенные сигналы. Каждый из них сигнализирует о том, что необходимо провести регуляцию численность виртуальных пользователей в системе. Регуляция численности пользователей осуществляется на следующем алгоритму. На Профиле нагрузки вызывается метод определения количество необходимых пользователей. Если число необходимых пользователей превышает количество текущих пользователей, тогда запускается механизм создания необходимого числа пользователей. Если число необходимых пользователей меньше числа текущих пользователей тогда запускается механизм за-

вершения работы необходимого числа пользователей. Прекращают свою работу наиболее старые пользователи.

За протоколированием всего происходящего во время тестирования отвечает внутренний механизм под названием Журнал. В Журнал поступает вся информация, которая была получена вовремя тестирования. Каждый этап тестирования записывается в журнал, чтобы потом было проще разобраться в причинах неудачного запуска теста.

Во время тестирования Генераторы пользователей сообщают информацию о текущем количестве пользователей и проводимой ими коррекции, сколько пользователей добавили сколько пользователей завершили свою работу. У каждого пользователя есть свой таймер времени, сколько он работает по завершению эта информация поступает в Журнал, также с меткой о том, почему завершилась работа. Закончилась по времени, из-за внезапной ошибки, автомат попал в тупик или поступил сигнал о завершении.

Вся информация о работе автоматов сохраняется в журнале. К такой информации относятся слова, которые поступили на вход, переходы, которые сработали и т.д. Эта информация крайне полезна для воспроизведения случаев, связанных с ошибками. Агрегация этой информации может показать движение пользователей в системе, это может быть полезно для определения соответствует ли поведение ожидаемому. При возникновении ошибок во время тестирования собирается так же stack trace (трассировка стека). Отдельно регистрируются случаи, когда состояние системы отличается от ожидаемого. В этом случаи сохраняется информация о том, по каким критерием выявлено несоответствие. Здесь возможно сохранять страницу целиком для дальнейшего анализа. Случаи, когда автомат завершает свою работу из-за того, что оказался в тупике регистрируются отдельно, так как такое поведение может быть совершенно нормальным, анализ таких случаев остается на усмотрение тестировщика.

Отдельно стоит отметить, что во время тестирования собирается информация о технических характеристиках системы, к такой информации относится величина передаваемых страниц, время передачи и т.д. Эта информация постоянно изменяется и на её основе вычисляются следующие значения среднее, медианное, минимальное и максимальное значение. Анализ этой информации позволяет понять, как ведет себя система во время работы. С учетом того, что задача тестирования создать условия максимально приближенные к реальным, то получаемые характеристики производительности системы, более точны чем у нагрузочного тестирования.

Для того, чтобы определить откуда поступило сообщение в журнал, генераторы передают свой уникальный идентификатор, который назначается им при создании. При создании пользователей генераторами им так же присваивается уникальный идентификатор и вместе с тем они получают идентификатор генератора, таким образом можно легко найти записи в журнале сделанные определенным пользователем. Стоит отметить, что во время выполнения пользовательских действий они так же могут писать свои собственные сообщения в журнал. Дополнительная информация может предоставлять более полезную и релевантную информацию при разборе журнала.

Для запуска тестов используется консольное приложение со своим интерфейсом командной строки (англ. Command Line Interface, CLI). Благодаря чему оно может быть интегрировано с другими инструментами начиная от графических оболочек и до Continuous Integration инструментов.

Для запуска тестирования необходимо указать путь до сборки и до файла настроек, в котором указывается продолжительность тестирования и набор пар имя Профиля пользователя и имя Профиля нагрузки. Под сборкой подразумевается набор скомпилированных тестов в виде одной библиотеке.

После запуска тестирования последовательно выполняются следующие фазы: загрузка сборки, инициализация Генераторов пользователей, пользовательская фаза инициализации, фаза тестирования, пользовательская фаза очистки, обработка результатов.

После завершения тестирования формируется отчет, который выводится на экран и сохраняется в отдельный файл. Отчет состоит нескольких разделов. Первый раздел содержит в себе информацию об ошибках, которые возникли во время тестирования и у какого пользователя. Под ошибками подразумевается все случаи, когда ожидаемое поведение системы отличалось от фактического. Фиксируются случаи перехода автомата в тупиковое

Аппаратно-программное обеспечение информационно-вычислительных систем и систем управления состояние. Информация из данного раздела акцентирует внимание на нарушении бизнес процессов в системе.

Следующий раздел посвящен характеристикам производительности веб сервера. Выводится минимальное, средние и максимальное время ответа от сервера, величина страниц, количество запросов отправляемые серверу. А также выводится информация по каждому конкретной странице. Информация из данного раздела акцентирует внимание на то как ведется себя веб-сервер во время нагрузки, и как влияет нагрузка системы на отдельные её узлы. Используя разные профили нагрузки можно имитировать различную нагрузки на разные участки системы, что будет давать более объективную картину системы.

Библиографический список

1. Альфа- и бета-тестирование. – URL: <http://training.qatestlab.com/front-page/blog/technical-articles/alpha-beta-testing/> (дата обращения: 21.04.2018).
2. Performance Testing Guidance for Web Applications / J. D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, and Dennis Rea // Microsoft Press; 1 edition. – November 21. – 2007. – 288 p.
3. Lee, D. Principles and methods of testing finite state machines / D. Lee, M. Yannakakis // A survey. Proceedings of the IEEE. – 1996. – Vol. 84(8). – P. 1090–1123.
4. Kovalenko, D. Selenium Design Patterns and Best Practices / D. Kovalenko. – Packt Publishing, 2014. – 270 p. – URL: <http://livelib.ru>
5. Meszaros, Gerard. XUnit Test Patterns: Refactoring Test Code / Gerard Meszaros // Addison Wesley; 01 edition. – May 21. – 2007. – 944 p.
6. Финаев, В. И. Аналитические и имитационные модели : учеб. пособие / В. И. Финаев, Е. Н. Павленко, Е. В.Заргарян. – Таганрог : Изд-во Технологического ин-та ЮФУ, 2007. – 310 с.
7. Миронов, А. М. Основные понятия теории вероятностных автоматов / А. М. Миронов // Интеллектуальные системы. Теория и приложения. – 2015. – Т. 18, вып. 4. – С. 267–291.
8. Миронов, А. М. Основные понятия теории вероятностных автоматов / А. М. Миронов // Интеллектуальные системы. Теория и приложения. – 2016. – Вып. 2, ч. 2. – URL: <http://istina.msu.ru>
9. Andy, Hunt Unit Testing in C# with NUnit, Pragmatic Bookshelf / Hunt Andy, Thomas Dave, Hargett Matt. – 2007. – 220 p. – URL: <http://books.google.kz>
10. Erinle, B. Performance Testing with JMeter, Packt / B. Erinle. – 2015. – 164 p. – URL: <http://twirpx.com>

Д. А. Хоцян, О. Э. Яремко

Пензенский государственный университет, Пенза, Россия

МЕТОД ФУНДАМЕНТАЛЬНЫХ РЕШЕНИЙ В ТЕОРИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ЭЛЛИПТИЧЕСКОГО ТИПА

Рассматривается метод фундаментальных решений для решения задачи Дирихле в верхней полуплоскости. Метод реализован в приложении Curve Fitting Toolbox среды MATLAB. График решения задачи и построение линий уровня решения задачи визуализированы при помощи графического интерфейса GUI MATLAB.

Наряду с сеточными методами, гораздо выгоднее применять бессеточные методы, например, метод радиальных базисных функций и, в частности, метод фундаментальных решений, как средства решения дифференциальных уравнений.

Основной принцип решения, применяя радиальные базисные функции, – аппроксимация неизвестного решения при помощи функции специального вида.

Преимуществом метода является формирование оптимальных параметров радиальных базисных функций, используя принципы обучения.

Задача Дирихле для верхней полуплоскости $\text{Im}(z) > 0$ состоит в том, чтобы найти функцию $\varphi(x, y)$, гармоническую в верхней полуплоскости и имеющую граничные значения $\varphi(x, 0) = U(x)$, где $U(x)$ является вещественной функцией вещественной переменной x [2].

Важным методом решения этой задачи является результат, который приписывается французскому математику Симеону Пуассону.

Теорема 1 – (интегральная формула Пуассона). Пусть U – вещественная функция, кусочно-непрерывная и ограниченная для всех вещественных t [1]. Функция

$$\varphi(x, y) = \frac{y}{\pi} \int_{-\infty}^{\infty} \frac{U(t)}{(x-t)^2 + y^2} dt$$

– гармоническая в верхней полуплоскости и имеет граничные значения $\varphi(x, 0) = U(x)$, $U(x)$ везде непрерывна.

Пример 1. Найти гармоническую функцию $\varphi(x, y)$ в верхней полуплоскости $\text{Im}(z) > 0$, которая принимает граничные значения

$$\begin{aligned} \varphi(x, 0) &= \arctg(x+1) + \arctg(1-x) + 0,1(\text{rand}(\text{size}(x)) - 0,5), \\ &\text{при } -\infty \leq x \leq \infty; \end{aligned}$$

Решение. Средствами Maple получаем решение:

$$\varphi(x, y) = \arctg\left(\frac{x+1}{y+1}\right) + \arctg\left(\frac{-x+1}{y+1}\right)$$

Применим метод фундаментальных решений. Аппроксимация решения ищется в виде одной обратной квадратики – фундаментального решения

$$u_{apr} = \left(\frac{a}{b}\right) \frac{(y+b)}{(y+b)^2 + (x-c)^2};$$

где a – вес, b – ширина окна, c – центр.

Для определения параметров фундаментального решения использовался пакет Curve Fitting Toolbox. Используя MATLAB, построим линии уровня решения задачи Дирихле и график решения задачи Дирихле.

Листинг 1. Результат работы программы, определяющей параметры фундаментального решения

General model :

$$gfit(x) = a * (b^2 + (x-c)^2)^{-1}$$

Coefficients (with 95% confidence bounds):

$$a = 2.781 (2.652, 2.909)$$

$$b = 1.297 (1.259, 1.335)$$

$$c = -0.003452 (-0.02883, 0.02192)$$

Листинг 2. Программа построения линий уровня решения

$$[x, y] = \text{meshgrid}(-4:0.1:4, 0:0.1:5);$$

$$ut = \text{atan}((x+1)./(y+1)) + \text{atan}((-x+1)./(y+1));$$

$$uapr = (a/b) * (y+b)./((y+b).^2 + (x-c).^2);$$

$$[x, y] = \text{meshgrid}(-4:0.1:4, 0:0.1:5);$$

$$\text{contour}(x, y, uapr, 30)$$

Листинг 3. Программа построения графика решения задачи Дирихле

$$[x, y] = \text{meshgrid}(-4:0.1:4, 0:0.1:5);$$

$$ut = \text{atan}((x+1)./(y+1)) + \text{atan}((-x+1)./(y+1));$$

$$uapr = (a / b^2) * (y + b^2) / ((y + b^2)^2 + (x - c)^2);$$

$$[x, y] = \text{meshgrid}(-4:0.1:4, 0:0.1:5);$$

$$\text{mesh}(x, y, (uapr));$$

Для визуализации использовали приложение GUI MATLAB, в котором были выбраны опции для построения 3D графика решения и линий уровня решения.

Результаты работы программы представлены на рис. 1, 2.

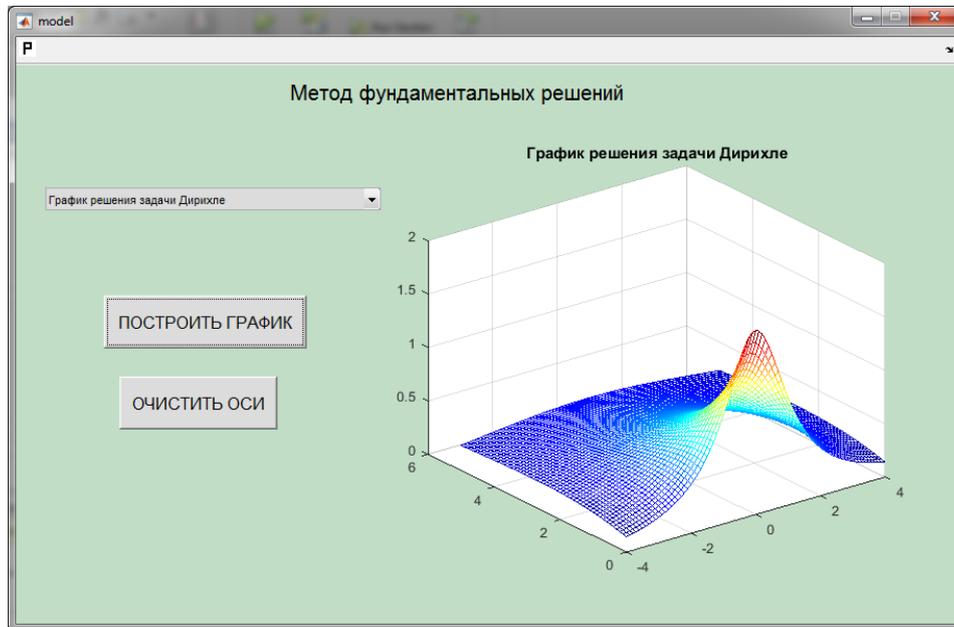


Рис. 1. График решения задачи Дирихле

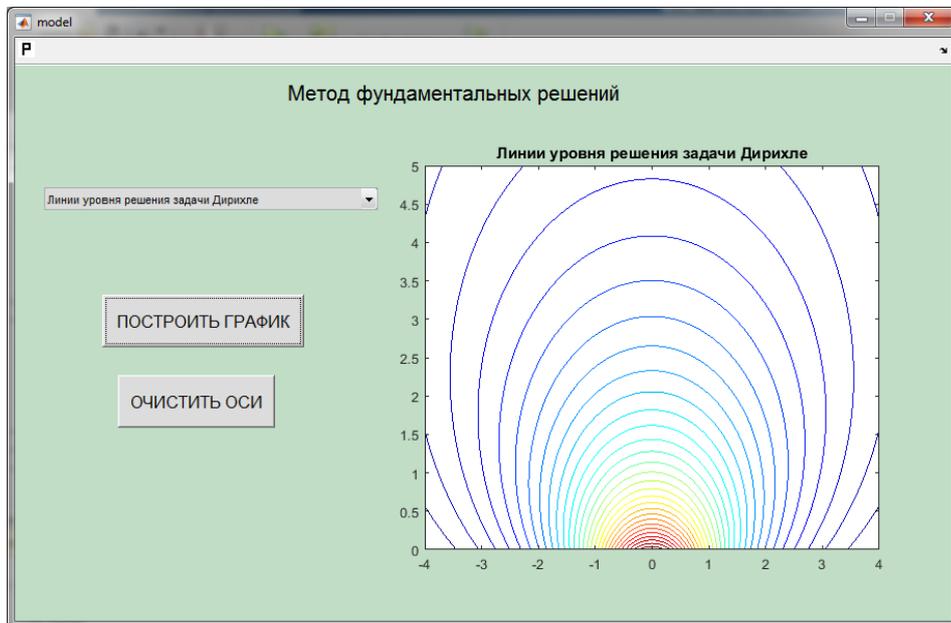


Рис. 2. График линий уровня решения задачи Дирихле

Из рис. 2 можно увидеть, что чем дальше линии уровня от нуля, тем меньше величина потенциального поля.

1. Снеддон, И. Н. Классическая теория упругости / И. Н. Снеддон, Д. С. Берри. – М. : Гос. изд-во физ.-мат. лит., 1961. – 219 с.
2. Яремко, О. Э. Метод радиально базисных функций в задаче Дирихле для уравнения Лапласа и его реализация в MATLAB / О. Э. Яремко, Д. А. Хоцян // Проблемы информатики в образовании, управлении, экономике и технике : сб. XVII Междунар. науч.-техн. конф. – Пенза : ПГУ, 2017. – 5 с.

Л. В. Тельнова, Н. Ю. Кудряшова

Пензенский государственный университет, Пенза, Россия

МЕТОД КВАЗИОБРАЩЕНИЯ ДЛЯ РЕШЕНИЯ СТАЦИОНАРНОЙ ГРАНИЧНОЙ ЗАДАЧИ ТЕПЛОПРОВОДНОСТИ

Исследуется задача восстановления температуры на поверхности стержня по результатам измерений на границах области.

Для активного развития авиации, усовершенствования космических аппаратов, тепловой защиты высокоскоростных летательных аппаратов, улучшения энергосистем для авиационной и ракетной техники необходимо непрерывно совершенствовать науку о теплообмене и развивать теорию теплопередачи. Проводятся теплофизические исследования и математическое моделирование, которые бывают связаны с решением обратных задач. При их изучении выделяют ретроспективную (отсутствуют начальные условия), граничные (не заданы необходимые граничные условия) и коэффициентные (исходная задача задана не точно) обратные задачи. Точное решение таких задач не всегда бывает возможность получить. В связи с этим возникает необходимость в разработке и исследовании численных методов их решения. В работе исследуется и реализуется численный метод решения стационарной граничной задачи, когда температурное поле не изменяется по времени, решение которой необходимо при изучении ряда важных практических задач, таких как прогрев теплозащитных оболочек летательных аппаратов, нагрев стенок сопел двигателей.

Будем рассматривать стационарное поле в некотором однородном образце, у которого прямоугольное сечение $\Omega = \{x | x = (x_1, x_2), 0 < x_\alpha < l_\alpha, \alpha = 1, 2\}$ [1].

Запишем уравнение Лапласа для данного случая

$$-\sum_{\alpha=1}^2 \frac{\partial^2 u}{\partial x_\alpha^2} = 0, \quad x \in \Omega. \quad (1)$$

Отметим часть границы $\gamma = \{x | x \in \partial\Omega, x_2 = l_2\}$, на которой прямые измерения невозможны (рис. 1):. Обозначим $\partial\Omega \setminus \gamma = \Gamma + \gamma_*$, а $\gamma_* = \{x | x \in \partial\Omega, x_\alpha = 0\}$.

Примем, что часть границы не проводит тепло, т.е. введем в рассмотрение дополнения, которые будут однородными граничными условиями второго рода:

$$\frac{\partial u}{\partial n} = 0, \quad x \in \Gamma + \gamma_*. \quad (2)$$

Поставленная задача исследуется в простейшей трактовке, когда будут выполнены дополнительные температурные измерения в области γ_* :

$$u(x) = g(x), \quad x \in \gamma_*. \quad (3)$$

В представленной задаче (1)–(3) в области γ_* известны температура и поток. Следовательно, трактовать эту задачу будем как задачу по продолжению температурного поля с области γ_* во всю Ω , т.е. рассматриваем эволюционную задачу по переменной x_2 . Эво-

Аппаратно-программное обеспечение информационно-вычислительных систем и систем управления люционную составляющую обозначим t и положим $x = x_1$. Значит уравнение (1) можно переписать следующим образом

$$\frac{\partial^2 u}{\partial t^2} - Lu = 0, \quad 0 < t \leq T, \quad 0 < x < l, \quad (4)$$

где $l = l_1$, $T = l_2$, а

$$Lu = -\frac{\partial^2 u}{\partial x^2}. \quad (5)$$

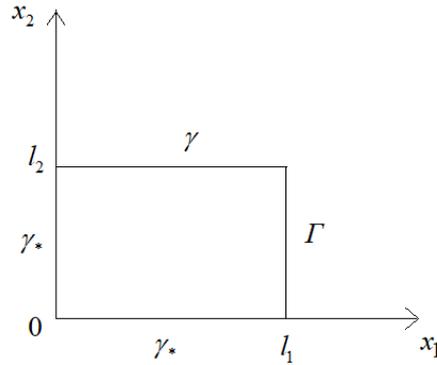


Рис. 1. Сечение стержня

Условия на границе (2), (3) перепишем в виде

$$\frac{\partial u}{\partial x}(x, t) = 0, \quad x = 0, l, \quad (6)$$

$$u(x, 0) = u^0(x), \quad 0 < x < l, \quad (7)$$

$$\frac{\partial u}{\partial t}(x, 0) = 0, \quad 0 < x < l. \quad (8)$$

Рассмотрим метод квазиобращения для получения решения задачи (4)–(8). Приближенное решение этой задачи обозначим за $u_\alpha(x, t)$ и определим из равенства

$$\frac{\partial^2 u_\alpha}{\partial t^2} - Lu_\alpha + \alpha L^2 u_\alpha = 0, \quad 0 < t \leq T, \quad 0 < x < l, \quad (9)$$

с условиями

$$\frac{\partial u_\alpha}{\partial x}(x, t) = 0, \quad x = 0, l, \quad (10)$$

$$L \frac{\partial u_\alpha}{\partial x}(x, t) = 0, \quad x = 0, l, \quad (11)$$

$$u_\alpha(x, 0) = u^0(x), \quad 0 < x < l, \quad (12)$$

$$\frac{\partial u_\alpha}{\partial t}(x, 0) = 0, \quad 0 < x < l. \quad (13)$$

Рассмотрим разностную схему для изученного выше варианта метода квазиобращения [2]:

$$B \frac{y_{n+1} - y_{n-1}}{2\tau} + \tau^2 R \frac{y_{n+1} - 2y_n + y_{n-1}}{\tau^2} + Ay_n = 0, \quad n = 1, 2, \dots \quad (14)$$

Обозначим

$$\Lambda v = \begin{cases} \frac{2}{h} v_{\bar{x}}, & x = l, \\ -v_{\bar{x}\bar{x}}, & x \in \omega, \\ -\frac{2}{h} v_x, & x = 0 \end{cases} \quad (15)$$

и получим соответствующую дифференциально-разностную задачу:

$$\frac{\partial^2 v}{\partial t^2} - \Lambda v + \alpha \Lambda \frac{\partial^2 v}{\partial t^2} = 0, \quad x \in \bar{\omega}, \quad 0 < t \leq T, \quad (16)$$

$$v(x, 0) = u^0(x), \quad x \in \bar{\omega}, \quad (17)$$

$$\frac{\partial v}{\partial t}(x, 0) = 0, \quad x \in \bar{\omega}. \quad (18)$$

Тогда трехслойную разностную схему с весами (16) – (18) запишем так:

$$(E + \alpha \Lambda) \frac{y_{n+1} - 2y_n + y_{n-1}}{\tau^2} - \Lambda(\sigma_1 y_{n+1} + (1 - \sigma_1 - \sigma_2)y_n + \sigma_2 y_{n-1}) = 0, \quad (19)$$

$$n = 1, 2, \dots$$

при известных y_1 и y_0 (пусть $y_0 = y_1 = u^0(x), x \in \bar{\omega}$).

Схему (19) будем решать методом разностной прогонки. Разностную краевую задачу запишем в виде [3]

$$A_i y_{i-1} - C_i y_i + B_i y_{i+1} = F_i, \quad A_i \neq 0, B_i \neq 0, i = 1, 2, \dots, N-1, \quad (20)$$

$$y_0 = \lambda_1 y_1 + \mu_1, \quad y_N = \lambda_2 y_{N-1} + \mu_2$$

Применим алгоритм прогонки к (19)

$$(E + \alpha \Lambda) \frac{y_{n+1} - 2y_n + y_{n-1}}{\tau^2} - \Lambda(\sigma_1 y_{n+1} + (1 - \sigma_1 - \sigma_2)y_n + \sigma_2 y_{n-1}) = 0,$$

здесь

$$\Lambda y_n = \frac{y_{m+1}^n - 2y_m^n + y_{m-1}^n}{\tau^2}, \quad \text{где } y_m^n = y(x_m, t_n);$$

$$\Lambda \frac{y_m^{n+1} - 2y_m^n + y_m^{n-1}}{\tau^2} = \frac{\Lambda y_m^{n+1} - 2\Lambda y_m^n + \Lambda y_m^{n-1}}{\tau^2}.$$

Тогда схема (19) примет вид

$$\frac{y_m^{n+1} - 2y_m^n + y_m^{n-1}}{\tau^2} + \frac{\alpha}{\tau^2} \left(\frac{y_{m+1}^{n+1} - 2y_m^{n+1} + y_{m-1}^{n+1}}{h^2} - 2 \frac{y_{m+1}^n - 2y_m^n + y_{m-1}^n}{h^2} + \frac{y_{m+1}^{n-1} - 2y_m^{n-1} + y_{m-1}^{n-1}}{h^2} \right) - \left(\sigma_1 \frac{y_{m+1}^{n+1} - 2y_m^{n+1} + y_{m-1}^{n+1}}{h^2} + (1 - \sigma_1 - \sigma_2) \frac{y_{m+1}^n - 2y_m^n + y_{m-1}^n}{h^2} + \sigma_2 \frac{y_{m+1}^{n-1} - 2y_m^{n-1} + y_{m-1}^{n-1}}{h^2} \right) = 0, \quad (21)$$

начальными условиями

$$y_0 = y_1 = u^0(x);$$

$$\frac{y_1 - y_0}{\tau} = 0.$$

Представим (21) в виде СЛАУ:

$$y_m^{n+1} + r \left(\frac{\alpha}{\tau^2} - \sigma_1 \right) (y_{m+1}^{n+1} - 2y_m^{n+1} + y_{m-1}^{n+1}) = 2y_m^n + r \left(\frac{2\alpha}{\tau^2} - 1 + \sigma_1 + \sigma_2 \right) (y_{m+1}^n - 2y_m^n + y_{m-1}^n) - y_m^{n-1} - r \left(\frac{\alpha}{\tau^2} - \sigma_2 \right) (y_{m+1}^{n-1} - 2y_m^{n-1} + y_{m-1}^{n-1})$$

Тогда если расписать задачу в виде (1.5.1), то ее коэффициенты определяются следующим образом:

$$A_i u_{i-1} - C_i u_i + B_i u_{i+1} = F_i, \quad A_i = r \left(\frac{\alpha}{\tau^2} - \sigma_1 \right), \quad B_i = r \left(\frac{\alpha}{\tau^2} - \sigma_2 \right), \quad C_i = 2r \left(\frac{\alpha}{\tau^2} - \sigma_1 \right) - 1,$$

$$F_i = 2y_m^n + r \left(\frac{2\alpha}{\tau^2} - 1 + \sigma_1 + \sigma_2 \right) \left(y_{m+1}^n - 2y_m^n + y_{m-1}^n \right) - \\ - y_m^{n-1} - r \left(\frac{\alpha}{\tau^2} - \sigma_2 \right) \left(y_{m+1}^{n-1} - 2y_m^{n-1} + y_{m-1}^{n-1} \right), \quad y_n^0 = 0 = \mu_1, \quad y_n^M = 0 = \mu_2, \quad \lambda_1 = 0 = \lambda_2.$$

Применим метод квазиобращения к решению следующей задачи:

$$-\frac{\partial^2 u}{\partial x_1^2} - \frac{\partial^2 u}{\partial x_2^2} = 0, \quad 0 < x_1 \leq 1, \quad 0 < x_2 < 1, \\ \frac{\partial u}{\partial n} = 0, \quad x \in \Gamma + \gamma_*, \quad u(x) = x^2, \quad x \in \gamma_*. \quad (22)$$

Численная реализация задачи (22) представлена в табл. 1.

В данной работе был исследован и реализован метод квазиобращения решения обратной граничной задачи теплообмена в случае, когда температурное поле не изменяется с течением времени.

Таблица 1

| x_1 | y_{2h} | y_h | $ y_{2h} - y_h $ |
|--|----------|----------|------------------|
| $m = 5000, n = 5000, \alpha = 0.5, \sigma_1 = 0.5, \sigma_2 = 0.5$ | | | |
| (0.40008;0,00020004) | 0.600241 | 0.600245 | 3.2135e-06 |
| (0,80016; 0,00020004) | 0.570086 | 0.570082 | 1.1078e-06 |
| (0.40008;0,40028) | 0.490131 | 0.490069 | 6.195e-05 |
| (0,80016; 0,00020004) | 0.951472 | 0.951415 | 2.66794e-05 |
| (0,40008;0,80036) | 0.799992 | 0.799746 | 0.000245795 |
| (0,80016; 0,80036) | 1.46293 | 1.46275 | 0.00018239 |
| $m = 5000, n = 5000, \alpha = 0.5, \sigma_1 = 0.01, \sigma_2 = 0.99$ | | | |
| (0.40008;0,00020004) | 0.300512 | 0.300518 | 5.0337e-06 |
| (0,80016; 0,00020004) | 0.270047 | 0.270043 | 3.5623e-06 |
| (0.40008;0,40028) | 0.490128 | 0.490077 | 5.0882e-05 |
| (0,80016; 0,00020004) | 0.951471 | 0.951389 | 8.1646e-05 |
| (0,40008;0,80036) | 0.799954 | 0.799777 | 0.00017737 |
| (0,80016; 0,80036) | 1.46288 | 1.46261 | 0.000277149 |
| $m = 5000, n = 5000, \alpha = 0.25, \sigma_1 = 0.5, \sigma_2 = 0.5$ | | | |
| (0.40008;0,00020004) | 0.50013 | 0.50018 | 1.3456e-05 |
| (0,80016; 0,00020004) | 0.91012 | 0.91016 | 1.7896e-05 |
| (0.40008;0,40028) | 0.678673 | 0.678512 | 0.000161737 |
| (0,80016; 0,00020004) | 1.18007 | 1.17992 | 0.00015199 |
| (0.40008;0,80036) | 1.87465 | 1.87383 | 0.000814844 |
| (0,80016; 0,80036) | 2.69809 | 2.69747 | 0.000626018 |

Решение модельных задач показало эффективность рассмотренных вычислительных алгоритмов.

Библиографический список

1. Самарский, А. А. Вычислительная теплопередача / А. А. Самарский, П. Н. Вабищевич. – М. : Едиториал УРСС, 2003. – 784 с.
2. Цирельман, Н. М. Прямые и обратные задачи тепломассопереноса / Н. М. Цирельман. – М. : Энергоатомиздат, 2005. – 392 с.
3. Тельнова, Л. В. Обратные граничные задачи теплопроводности / Л. В. Тельнова, Н. Ю. Кудряшова // Информационные технологии в науке и образовании. Проблемы и перспективы : V Ежегодная межвуз. студ. науч.-практ. конф. (14 марта 2018 г.). – Пенза : Изд-во ПГУ, 2018. – URL: <http://fvt.pnzgu.ru>

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА ПРОГРАММ ОБЕСПЕЧЕНИЯ МОДЕЛИРОВАНИЯ СИСТЕМ СЕТЯМИ ПЕТРИ

Рассматривается программная реализация приложения для моделирования систем сетями Петри. Приводится программа, реализующая построение и анализ сетей Петри.

В последнее время с увеличением сложности дискретных систем все большее распространение получают «неалгоритмические» параллельные системы с недетерминированным поведением, в которых каждая компонента функционирует, как правило, независимо, взаимодействуя с другими компонентами время от времени. Эти системы с асинхронно взаимодействующими и параллельно функционирующими элементами не были описаны в терминах классической теории автоматов. Так как такие понятия, как состояние автомата и глобальная функция перехода, не очень удобны для эффективной и экономной характеристики недетерминированной динамики поведения системы с локальными связями между независимыми параллельными процессами.

Из множества предложенных методов описания и обработки дискретных параллельных систем выделяется метод, предложенный Карлом Петри для моделирования поведения асинхронных потоков в системе преобразования данных, который был основан на сетевых моделях.

Сети Петри представляют собой удобный, широко используемый и мощный инструмент для исследования моделируемых систем. Такой инструмент обладает наилучшими возможностями для описания взаимосвязей и взаимодействий работающих процессов [3].

Сегодня сети Петри становятся все более используемыми, так как могут применяться для решения множества задач:

- моделирования взаимосвязанных и параллельных процессов;
- моделирования работы телефонных систем и анализа ее свойств;
- разработка параллельных графических объектно-ориентированных систем программирования;
- разработка программного обеспечения центров для дистанционного контроля и управления.

Исходя из вышесказанного, можно заключить, что сети Петри используются для описания и анализа динамических систем [1].

Очень часто динамические системы бывают очень сложными, включают в себя большое количество взаимодействующих компонент, где каждая из них также может представлять сложную динамическую систему, взаимодействующую с другими элементами системы.

В связи с этим, чтобы снизить трудозатраты при проведении проектных работ, снизить время, затрачиваемое на проектирование систем и разработку устройств, было разработано программное средство, позволяющее разрабатывать и анализировать модели систем с помощью аппарата сетей Петри. Разработанная система дает возможность строить модели систем с помощью сетей Петри, а также анализировать динамику их выполнения.

Продемонстрируем пример работы приложения на примере моделирования системы «Протокола передачи данных» (рис. 2). Эта система была выбрана как общеизвестный пример, с помощью которого можно проиллюстрировать в рамках представленной работы возможности приложения, хотя предлагаемый способ проверки достижимости применим и к более крупным системам.

Данная система может использоваться для передачи данных между передатчиком и приемником, т.е. передатчик может передавать необходимое количество сообщений приемнику. Моделируемая система имеет в своем составе три основных потока: первый – отправление сообщений от источника к приемнику, второй – подтверждение доставки сообщения источником приемнику, третий – запрос приемника для получения данных от отправителя. Таким образом, передатчик отправляет информацию получателю и принимает сообщения о ее доставке. В приемник поступают отправленные данные, после чего он посылает сообщения о получении. Данная сеть (см. рис. 1) состоит из 14 мест P, 8 переходов T.

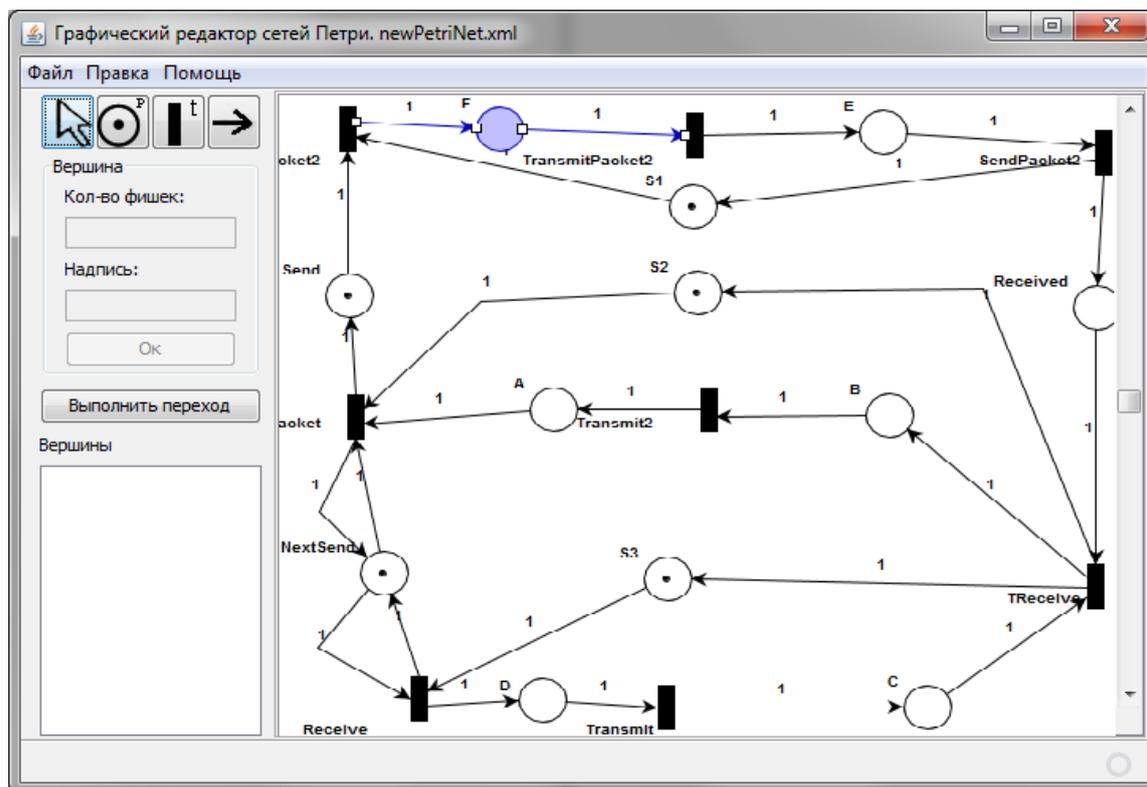


Рис. 1. Сеть Петри «протокол передачи данных»

В программе также реализована возможность анализа сети Петри на основе дерева достижимости. Данный метод возможно использовать для исследования покрываемости, сохраняемости, безопасности и ограниченности сетей Петри [2]. Пример построенного дерева достижимости можно увидеть на рис. 2.

Также был реализован метод приведения сети Петри к матричному виду. Большим плюсом матричного представления сетей Петри является то, что такое представление более удобно с точки зрения понимания цифровой информации компьютерными системами и различными приложениями. Это дает возможность анализа получившихся матриц в пакетах прикладных программ для решения задач технических вычислений и анализа данных (таких как Matlab, MathCad). Так как такие приложения обеспечивают эффективный анализ матричных данных, добавлена функция сохранения созданных сетей Петри в матричный вид, для возможности переноса сети Петри в виде матрицы, что обеспечивает переносимость созданной модели на другие системы анализа.

На примере построенной модели «Протокола передачи данных» была продемонстрирована работа приложения. На основе проделанных действий было выяснено, что система работает без сбоев, корректно выполняя переходы в сети. Также система корректно отображает граф достижимости построенной сети Петри, и позволяет получить матричное представление сети Петри.

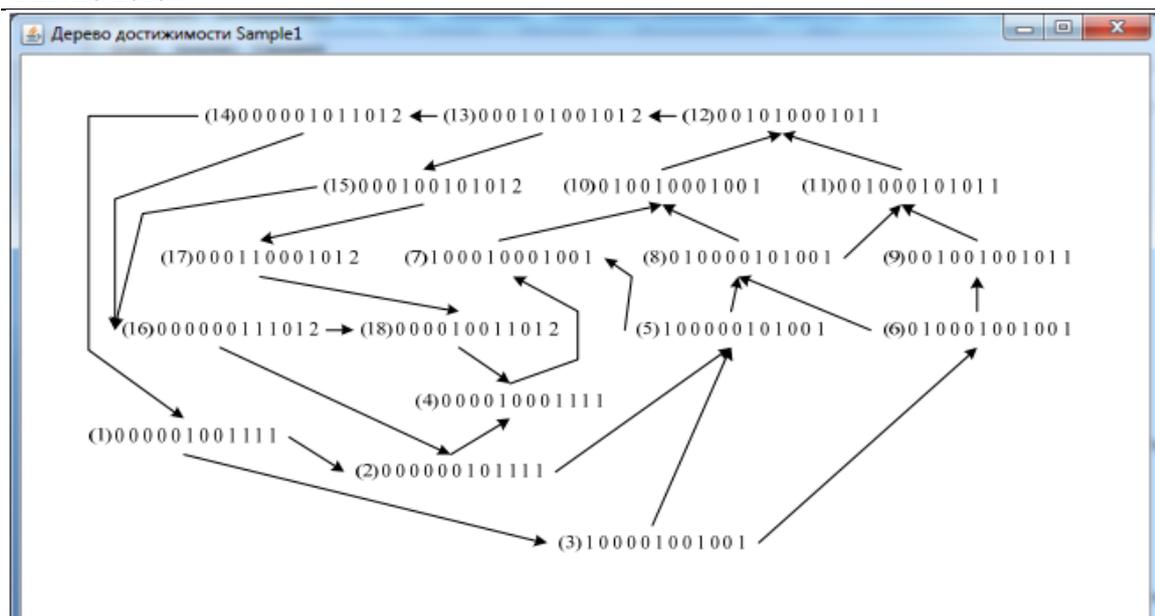


Рис. 2. Граф сети Петри

Библиографический список

1. Кирьяк, Б. А. Оптимизация быстродействия программ на основе сетей Петри / Б. А. Кирьяк // Прикладная математика и информатика: современные исследования в области естественных и технических наук : III науч.-практ. всерос. конф. молодых ученых. – Тольятти, 2017. – 6 с.
2. Кирьяк, Б. А. Программная поддержка сетей Петри / Б. А. Кирьяк // Информационные технологии в науке и образовании. Проблемы и перспективы : V Ежегодная межвуз. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2018. – 5 с.
3. Котов, В. Е. Сети Петри / В. Е. Котов. – М. : Наука, 1984. – 160 с.

Г. В. Бобрышева, А. О. Звозникова

Пензенский государственный университет, Пенза, Россия

ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ НА ПРЕДПРИЯТИИ

Приоритетной задачей при построении и использовании автоматизированных информационных систем на предприятии является защита информации от случайных и преднамеренных угроз. Решение задачи достигается за счет создания системы безопасности и непрерывного мониторинга ее эффективности с использованием качественных и количественных методов оценки информационной безопасности.

В настоящее время наблюдается бурное развитие и внедрение во все сферы производства автоматизированных информационных систем (АИС). Современные АИС хранят, обрабатывают и передают по каналам связи большое количество разнообразной информации, которая является часто конфиденциальной или содержит в себе персональные данные. Поэтому при построении и использовании АИС вопросам обеспечения информационной безопасности (ИБ) уделяется особое внимание и, в частности, вопросам защиты информа-

Аппаратно-программное обеспечение информационно-вычислительных систем и систем управления
ции от случайных и преднамеренных угроз, следствием которых является утечка, разрушение и модификация информации.

Вопросы обеспечения информационной безопасности направлены на решение проблем безопасности, которые могут появиться на разных стадиях жизненного цикла АИС и зависят от множества факторов [1].

Решение вопросов обеспечения информационной безопасности на предприятиях при использовании АИС достигается за счет создания системы безопасности, основной целью которой является предупреждение нарушений, которые могут возникнуть вследствие реализации случайных и преднамеренных угроз.

Основными факторами для реализации случайных угроз являются:

- ошибки проектирования;
- программные ошибки;
- схемные и системотехнические ошибки разработчиков;
- ошибки человека, например, оператора при вводе информации с терминала;
- помехи в каналах связи;
- чрезмерная сложность системы;
- недостаточность обучения;
- аварийные ситуации.

Основными факторами для реализации преднамеренных угроз являются:

- определенное недовольство человека своими жизненными условиями;
- материальный интерес;
- простое развлечение с самоутверждением своих способностей, например, как у хакеров.

Эффективность функционирующей на предприятии системы безопасности во многом определяется поддерживаемым уровнем информационной безопасности АИС и зависит от используемых методов и средств защиты информации.

Для оценки эффективности система безопасности применяют качественные и количественные методы оценки ИБ. Качественные методы оценки ИБ предполагают оценку уровня ИБ, анализ рисков и тестирование системы информационной безопасности (СИБ). К количественным методам относится метод экспертных оценок, метод информационных потоков, графовый метод и метод весовых коэффициентов.

Оценка уровня ИБ позволяет на основе некоторого набора исходных данных выявить качественное соответствие или несоответствие СИБ определенным требованиям.

Анализ рисков проводится для выбора подхода к управлению рисками ИБ и является основополагающим при построении системы управления информационной безопасностью предприятия. Эффективность данного процесса определяется точностью и полнотой анализа и оценки факторов риска, а также эффективностью используемых механизмов принятия управленческих решений и контроля их исполнения. Для оценки рисков существуют различные подходы. При этом выбор подхода зависит от уровня предъявляемых требований к уровню ИБ, характера потенциальных угроз и эффективности потенциальных механизмов защиты.

Тестирование системы информационной безопасности – это метод выявления недостатков в СИБ предприятия с точки зрения третьего лица (взломщика). Тестирование СИБ позволяет проверить и оценить эффективность используемых в ней механизмов защиты и их устойчивость к атакам, а также ее уязвимые места путем имитации атак, с которыми не удается справиться с помощью выбранных методов и средств защиты информации. [2]

В процессе тестирования возможна проверка следующих средств и методов защиты информации:

- алгоритмы запутывания;
- алгоритмы мутации;
- алгоритмы компрессии данных;
- алгоритмы шифрования данных;
- вычисление сложных математических выражений в процессе отработки механизма защиты;

- методы затруднения дизассемблирования;
- методы затруднения отладки;
- эмуляция процессоров и операционных систем;
- нестандартные методы работы с аппаратным обеспечением.

Для проверки методов и средств защиты информации в основном используют три подхода тестирования: по методу "черного ящика", по методу "белого ящика", тестирование на проникновение.

Использование метода "черного ящика" предполагает, что команда тестирования не знает конфигурацию и внутреннюю структуру объекта испытаний. В процессе тестирования используются все известные типы атак на СИБ, происходит эмуляция действий потенциальных злоумышленников, пытающихся взломать систему защиты.

Основным средством тестирования в данном случае являются сетевые сканеры, содержащие базу данных с описанием известных уязвимостей операционной системы, маршрутизаторов, сетевых служб и т.п., а также алгоритмов осуществления попыток вторжения.

Преимущества метода «черного ящика»:

- возможность выявления дефектов, которые невозможно найти методом «белого ящика», например, отсутствие некоторой функциональности СИБ;
- возможность подготовки тестовых данных и составления тест-кейсов сразу после подготовки документации на СИБ;
- тестирование проходит «с позиции пользователя».

Недостатки метода «черного ящика» заключаются в:

- необходимости большого количества тестов из-за множества тестовых данных, вследствие чего, часто тесты могут быть избыточными, так как неизвестна конфигурация и структура СИБ;
- сложности составления тестов при отсутствии полной документации на СИБ;
- возможности пропустить граничные значения при составлении тестовых данных из-за неочевидной документации.

Использование метода "белого ящика" предполагает, что команда тестирования, знает конфигурацию и внутреннюю структуру объекта испытаний. В процессе тестирования происходит проверка наличия и работоспособности механизмов безопасности СИБ, соответствия состава и конфигурации системы защиты требованиям безопасности и существующим рискам. Выводы о наличии уязвимостей делаются на основании анализа конфигурации используемых средств защиты и системного программного обеспечения, а затем проверяются на практике. В данном случае основным инструментом анализа являются программные средства анализа защищенности системного уровня. [2]

Преимущества метода «белого ящика»:

- знание внутренней логики и структуры кода помогает подготовить тестовые данные, которые позволяют эффективное тестирование СИБ;
- позволяет оптимизировать код путем удаления дополнительных строк кода, которые могут привести к дефектам в коде.

Недостатки метода «белого ящика»:

- требуются квалифицированные и опытные специалисты, так как знание кода и внутренней структуры являются необходимыми условиями для данного вида тестирования;
- почти невозможно проверить каждый кусок кода, чтобы выявить скрытые ошибки.

Тестирование на проникновение (тесты на преодоление защиты, penetration testing, pentest, пентест) заключается в санкционированной попытке обойти существующий комплекс средств защиты АИС. В ходе тестирования команда выполняет роль злоумышленника, мотивированного на нарушение информационной безопасности сети заказчика. Данный метод позволяет обнаружить недостатки защиты, которые не были учтены при выборе политики безопасности. [3]

Тестирование на проникновение системы может осуществляться с использованием следующих подходов:

- ручное тестирование;

– автоматическое тестирование, наиболее популярными инструментами для которого являются: Nmap, Nessus, Metasploit, Wireshark, OpenSSL, Cain & Abel, THC Hydra, w3af;

– сочетание ручного и автоматического тестирования, являющееся наиболее оптимальным подходом, содержащим плюсы двух предыдущих методов тестирования.

При проведении теста на проникновение используются общепризнанные стандарты и руководства по обеспечению информационной безопасности. [4-7]

- OWASP Testing Guide;
- OWASP Top10;
- Web Application Security Consortium Threat (WASC) Classification;
- стандарты серии ISO 17799/27000.

Каждая вновь созданная СИБ является вероятно защищенной, что связано с стремительным развитием информационных технологий и вычислительной техники, вследствие чего, каждый метод и средство защиты может оказаться в любой момент не эффективным. Поэтому тестирование СИБ должно осуществляться не только при создании СИБ, но и регулярно при ее использовании. При этом периодичность тестирования зависит от важности и ценности хранимой, обрабатываемой и передаваемой информации в АИС и должна устанавливаться руководством предприятия самостоятельно.

Результаты тестирования СИБ позволят руководству предприятия в принятии управленческих решений при организации защиты информации от несанкционированного доступа третьих лиц.

Библиографический список

1. Лаборатория качества. Что такое тестирование на проникновение – необходимость. – URL: <https://spark.ru/startup/antihacking/blog/7853/test-na-proniknovenie-neobhodimost> (дата обращения: 10.09.2018).
2. Полянский, Д. А. Оценка защищенности : учеб. пособие / Д. А. Полянский. – Владимир : Изд-во Владимир. гос. ун-та, 2005. – 80 с.
3. Лаборатория качества. Что такое тестирование на проникновение. – URL: <https://nauchforum.ru/studconf/science/9/35159> (дата обращения: 14.09.2018).
4. OWASP Testing Project. – URL: https://www.owasp.org/index.php/OWASP_Testing_Project (дата обращения: 16.09.2018).
5. OWASP Top Ten Project. – URL: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project (дата обращения: 16.09.2018).
6. The Web Application Security Consortium / Threat Classification. – URL: https://www.owasp.org/index.php/OWASP_Testing_Project (дата обращения: 17.09.2018).
7. Стандарты. ISO 27000 – Международные стандарты управления информационной безопасностью. Общие сведения о стандартах серии ISO 27000. – URL: <http://iso27000.ru/standarty/iso-27000-mezhdunarodnye-standarty-upravleniya-informacionnoi-bezopasnostyu-1/> (дата обращения: 12.09.2018).

А. В. Травянка, Г. В. Бобрышева, Г. О. Звозникова

Пензенский государственный университет, Пенза, Россия

ПРОГРАММНЫЕ РЕШЕНИЯ ДЛЯ ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

В настоящее время всех сферах деятельности большое внимание уделяется вопросам обеспечения безопасности ресурсов вычислительной сети от случайного или злонамеренного получения ее лицами или процессами, не имеющими на это полномочий. Эффективность решения данной проблемы во многом определяется

В настоящее время количество информации, хранимой, обрабатываемой и передаваемой по вычислительным сетям, непрерывно растет высокими темпами. При этом растут и требования к информации, а именно к ее достоверности, надежности и конфиденциальности. Поэтому во всех сферах деятельности большое внимание уделяется вопросам обеспечения безопасности ресурсов вычислительной сети от случайного или злонамеренного получения ее лицами или процессами, не имеющими на это полномочий. Следствием несанкционированного доступа к информации может быть нарушение ее физической и логической целостности, копирование, переадресация, кража и другие негативные последствия, которые могут привести к серьезным материальным убыткам.

В организациях проблема обеспечения безопасности информации является первоочередной задачей. Эффективность решения данной проблемы во многом определяется качеством оценивания возможных рисков информационной безопасности и управления ими.

Согласно ГОСТ Р ИСО/МЭК 27005–2010 риск информационной безопасности (information security risk) – это возможность того, что данная угроза сможет воспользоваться уязвимостью актива или группы активов и тем самым нанесет ущерб организации [1].

Анализ рисков информационной безопасности организации начинается с рассмотрения всех возможных потенциальных угроз с целью определить все возможные непредвиденные ситуации для их реализации и определить методы и средства защиты информации. Виды рисков информационной безопасности определяются в зависимости от источников их возникновения, способа реализации незаконного вторжения и цели. Известные риски информационной безопасности классифицированы с целью стандартизации методов и средств, применяемых для защиты информации. Основными классификационными признаками рисков информационной безопасности являются:

- физические источники;
- нецелесообразное пользование компьютерной сетью и Всемирной паутиной;
- утечка из закрытых источников;
- утечка техническими путями;
- несанкционированное вторжение;
- атака информационных активов;
- нарушение целостности модификации данных;
- чрезвычайные ситуации;
- правовые нарушения.

Анализ рисков информационной безопасности состоит в том, чтобы выявить существующие риски и дать им количественную оценку. Риск определяется вероятностью причинения ущерба и величиной ущерба, наносимого ресурсам вычислительной сети и, соответственно, организации в целом в случае осуществления угрозы безопасности. На основе результатов оценки уровня рисков определяются требования безопасности, методы и средства защиты ресурсов вычислительной сети.

Задача управления рисками информационной безопасности включает выбор и обоснование выбора контрмер, позволяющих снизить уровни рисков до приемлемой величины [2].

Анализ рисков и управление ими являются очень трудоемкими процессами, качество которых во многом определяется принятыми мерами по обследованию и минимизации рисков.

Для количественного и качественного анализа рисков информационной безопасности во многих организациях применяют специальные информационные технологии: RA2 art of risk, vsRisk, CRAMM, RiskWatch, ГРИФ, COBRA, MethodWare и др. [3–5].

Результаты аналитического анализа функциональных особенностей информационных технологий по оценке рисков информационной безопасности приведены в табл. 1.

Таблица 1

Оценка информационных технологий

| Критерий | CRAMM | RiskWatch | ГРИФ | Новый продукт |
|--|-------|-----------|------|---------------|
| Рассчитанность на организации разного размера и деятельности | + | + | + | + |
| Автоматизация «What-if» | - | + | - | + |
| Удобство восприятия графиков и отчетов | - | - | + | + |
| Простота использования | - | - | + | + |
| Бесплатное использование | - | - | - | - |
| Поддержка | + | + | + | + |
| Количественная оценка | + | + | + | + |
| Качественная оценка | + | - | + | + |
| Русская локализация | - | + | + | + |
| Повышение информированности сотрудников | - | - | - | + |
| Пригодность к регулярному использованию | + | + | + | + |
| Использование независимой оценки | + | + | - | + |
| Сетевое решение | - | - | + | + |

Результаты аналитического анализа позволили выделить основные недостатки рассмотренных информационных технологий, предназначенных для анализа рисков информационной безопасности:

- сложность использования;
- проблема с отображением символов кириллицы;
- неполный охват активов (большинство продуктов сосредотачиваются только на IT-активах);
- отсутствие средств построения модели активов;
- часть угроз не связана с соответствующими типами уязвимостей и категориями активов организации;
- невозможность добавления пояснений и обоснований выбора тех или иных значений вероятности угрозы и величины уязвимости, в результате чего при анализе результатов вероятности невозможно определить, почему были выбраны те или иные значения;
- описание механизмов контроля включает в себя только название и цели;
- неполная совместимость с международными стандартами;
- затруднение процесса осознания рисков, так как расчет рисков выполняется автоматически и скрыт от пользователя;
- высокая цена реализации.

Минимизация выделенных недостатков возможна за счет разработки новой информационной технологии, позволившей повысить качество анализа и управления рисками информационной безопасности и ориентированная на

- простоту использования: интуитивно понятный интерфейс, наличие комментариев и подсказок на каждом шаге работы;
- наличие русскоязычного и англоязычного интерфейса;
- соответствие международным стандартам;
- возможность добавления и редактирования пояснений и обоснований выбора тех или иных значений вероятности угрозы и величины уязвимости, а также описаний механизмов контроля;
- умеренную стоимость программного продукта (в сравнении с аналогичными программными продуктами);
- предоставление отчетной информации в виде электронных таблиц бесплатного программного продукта OpenOffice 3.3;

- бесплатное обучение сотрудников и сопровождение программного продукта для качественного и количественного анализа рисков в рамках лицензионного договора;
- выгрузку отчетных данных во внешние системы.

Библиографический список

1. ГОСТ Р ИСО/МЭК 27005–2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности (взамен: ГОСТ Р ИСО/МЭК ТО 13335-3–2007 и ГОСТ Р ИСО/МЭК ТО 13335-4–2007. Дата введения: 01.12.2011). – URL: <http://gostedu.ru/50679.html> (дата обращения: 15.09.2018).
2. Анализ и управление рисками – общие положения.– URL: <http://www.iso27000.ru/informacionnyie-rubriki/upravlenie-riskami/analiz-i-upravlenie-riskami> (дата обращения: 10.09.2018).
3. Методики оценивания рисков и их программные реализации в компьютерных сетях. – URL: <https://moluch.ru/archive/49/6279/> (дата обращения: 11.09.2018).
4. Методы и средства анализа рисков и управления ими в ИС. – URL: <https://www.bytemag.ru/articles/detail.php?ID=9076> (дата обращения: 12.09.2018).
5. Программные продукты для анализа рисков. – URL: <http://www.iso27000.ru/informacionnyie-rubriki/upravlenie-riskami/programmnyie-produkty-dlya-analiza-riskov> (дата обращения: 14.09.2018).

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ И МАШИННОЕ ОБУЧЕНИЕ

И. Ю. Шполянская, А. Р. Загиров

Ростовский государственный экономический университет, Ростов-на-Дону, Россия

МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ В СОЗДАНИИ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ

Рассматриваются возможности использования нового подхода в управлении веб-ориентированными информационными системами на основе обработки данных о поведении пользователей системы с помощью процедур машинного обучения. Предлагаемый подход позволит оптимизировать работу рекомендательной системы в условиях неполных данных на основе обучающих алгоритмов, формирующих в предложения пользователей в режиме онлайн.

С повсеместным распространением web-технологий и созданием web-ориентированных информационных систем в разных предметных областях – электронной коммерции, Web CRM систем, систем электронного обучения, вызывает необходимость учета информационных потребностей пользователя для последующей оптимизации взаимодействия с ним. Построение web-ориентированной информационной системы, которая будет адекватно реагировать на индивидуальные предпочтения каждого пользователя и в соответствии с этой информацией формировать персонализированные предложения потребителю является сложной задачей. Решение ее предусматривает использование множества различных технологий, в том числе методов интеллектуального анализа данных, коллаборативной фильтрации и машинного обучения. В целях повышения эффективности взаимодействия с пользователями в web-ориентированных системах разрабатываются процедуры анализа и предсказания поведения пользователя на сайте, выполняется персонализация сайта, разрабатываются адаптивные стратегии взаимодействия с пользователем в соответствии с его интересами и предпочтениями [1]. Одним из направлений персонализации web-приложений для взаимодействия с пользователями являются рекомендательные системы. Так, в электронном бизнесе для интернет-магазинов создаются системы рекомендаций различных групп товаров, в том числе видео, музыкальных произведений, мобильных приложений, и т.д. В области онлайн обучения рекомендательные системы определяют наиболее предпочтительные для конкретного обучающегося электронные курсы. В электронных системах подбора персонала формируются рекомендации для выбора наиболее подходящего кандидата. Существуют web-приложения со встроенной системой рекомендаций туристических маршрутов в соответствии с интересами и предпочтениями конкретных пользователей.

Рекомендательная система представляет собой набор сервисов и программных компонентов, используемых для определения потребностей пользователя, которые впоследствии позволят сформировать персонифицированные рекомендации. Данные сервисы улучшают способы взаимодействия между пользователем и web-приложением, так как посетитель получает возможность интерактивного взаимодействия с системой, вместо просмотра статической информации. Рекомендательные системы создают рекомендации, независимо для каждого конкретного пользователя, основываясь на его предыдущем поиске и совершённых покупках. Кроме того, при формировании рекомендаций учитывается поведение всех клиентов со сходными интересами и предпочтениями, которые фиксируются и анализируются системой на основе всех действий пользователей на сайте.

В целом все рекомендательные системы можно разделить на следующие три категории: основанные на контенте; коллаборативные; гибридные [2]. В рекомендательных системах, основанных на контенте вывод о полезности товара для потребителя делается, ис-

ходя из полезности, рейтинга, присвоенного потребителем сходным товарам. Например, чтобы рекомендовать тот или иной фильм потребителю, контентная рекомендательная система пробует отыскать сходство между фильмами, которым ранее пользователь поставил высокую оценку (общие актеры, режиссеры, жанры и т.д.). После этого пользователю будут рекомендованы фильмы, обладающие высокой степенью общности с предпочтениями потребителя. Контентный подход в рекомендательных системах во многом использует методы информационного поиска и фильтрации данных. Соответственно, большинство современных контентных систем работают с рекомендациями товаров, содержащих текстовую информацию.

Рекомендательные системы, основанные на коллаборативной фильтрации определяют полезность товара для каждого потребителя, основываясь на оценках, предоставленных ранее другими пользователями. Полезность товара для пользователя определяется на основе оценок пользователей, которые по своим характеристикам или предпочтениям схожи с данным пользователем. Некоторые из рекомендательных систем могут использовать гибридные методы, комбинирующие контентные и коллаборативные подходы, что позволяет избежать ограничений, свойственных каждой системе [3].

В рекомендательных системах используются различные методы и алгоритмы, в зависимости от специфики решаемой проблемы, и от специфики используемых данных. Многие из них относятся к области машинного обучения. Машинное обучение является разделом искусственного интеллекта, основной идеей которого является использование компьютером не ранее записанного алгоритма, а самостоятельное обучение. Большинство методов машинного обучения разрабатывалось в виде альтернативы традиционным статистическим подходам. Многие методы тесно связаны с извлечением информации на основе технологий интеллектуального анализа данных (data mining). В работе машинного обучения применяется множество различных алгоритмов. Часть из них достаточно специфичны, но существуют и универсальные, например, метод опорных векторов, бустинг над решающими деревьями и т.д. К универсальным алгоритмам можно отнести и нейронные сети.

Рассмотрим возможность использования методов машинного обучения в рекомендательных системах. Основными задачами, решаемыми в рекомендательных системах, являются задачи кластеризации и классификации. Процедуры кластеризации в рекомендательных системах используются для выделения групп схожих объектов – групп пользователей со схожими предпочтениями и групп продуктов со схожими характеристиками. Сходство между элементами (пользователями) вычисляется с помощью заранее определенных метрик близости объектов в пространстве признаков. В качестве признаков выступают те элементы, по которым совпадают в определенной степени интересы пользователей (для кинопорталов – фильмы и их характеристики, для музыкальных ресурсов это треки, для интернет-магазинов купленные продукты, и т.д.). Похожие по характеристикам пользователи объединяются в так называемые кластеры.

Существует большое количество различных алгоритмов кластеризации. Наиболее распространенным является алгоритм k-средних (k-means), который разделяет элементы на k кластеров [4]. Процедуры классификации позволяют соотнести конкретный объект с заранее предопределенным классом, группой (пользователей, продуктов). Наиболее часто в классической постановке эти задачи решаются известными методами Data Mining.

Рассмотрим возможность использования нейросетевых технологий при решении задач рекомендательной системы. С их помощью можно создавать довольно сложные модели, решать задачи классификации и регрессии, распознавания образов и прогнозирования [5]. С помощью нейронных сетей в результате их обучения появляется возможность выявления сложных зависимостей между исходными данными и результатами. Наиболее распространенной, в то же время, способной решать определенный круг задач, является топология сети типа «перцептрон». В задачах классификации часто используется многослойный перцептрон, с одним или несколькими скрытыми слоями, который достаточно прост в реализации и обучении. Сеть типа многослойного перцептрона обучается на основе алгоритмов обратного распространения ошибки или других моделей обучения с учителем. Использование нейронных сетей в решении задач кластеризации пользователей или пред-

ложений в рекомендательных системах может быть выполнено на основе методов обучения без учителя, путем построения сетей Кохоннена или автоэнкодера. В случае автоэнкодера нейронами входного и выходного слоев выступают одни и те же характеристики объектов, а формируемый скрытый слой с существенно меньшим числом нейронов будет определять своими весами принадлежность к тому или иному классу.

В качестве примера рассмотрим классификацию пользователей онлайн кинотеатра (интернет-магазина по продаже электронных версий кинофильмов) с помощью нейронной сети на основе анализа набора данных пользователя. Необходимо выяснить принадлежность зарегистрированного пользователя к одной или нескольким группам, которые определяются предпочтениями пользователей в выборе жанров кинофильмов: 1) группе любителей триллеров и детективов; 2) группе любителей комедийных и приключенческих фильмов; 3) группе любителей мелодрам. Из множества возможных характеристик пользователей, выберем набор характеристик, в соответствии с которым будет произведена классификация, разбиение пользователей на группы. Были выделены следующие характеристики: Возраст; Пол; Уровень образования; Тип характера; Профессиональная область деятельности. Эти данные о пользователе можно собрать на сайте путём проведения анкетирования. Пример анкеты представлен на рис. 1.

The image shows a web browser window with a survey form titled "Анкета". The form contains several input fields:

- "Укажите ваш пол" (Specify your gender) with radio buttons for "Мужской" (Male) and "Женский" (Female).
- "Ваш возраст" (Your age) with a date picker set to "15 May 2013".
- "Выберите вид вашего образования" (Select your education level) with a dropdown menu showing "Высшее" (Higher).
- "Выберите тип вашего характера" (Select your character type) with a dropdown menu showing "Холерик" (Choleric).
- "Укажите область вашей профессиональной деятельности" (Specify your professional activity) with a dropdown menu showing "Строительство" (Construction).

Рис. 1. Пример анкеты пользователя

После сбора данных важная задача – препроцессинг, отображение данных в формат, пригодный для построения и обучения модели. Одним из основных способов построения векторного пространства признаков является представление категориальных признаков, как вектора в векторном пространстве с размерностью, соответствующей количеству возможных категорий. При этом значение координаты этой категории берется за единицу, а все остальные координаты обнуляются. В соответствии с такой кодировкой задаются характеристики Уровень образования, Тип характера, Профессиональная область деятельности. На рис. 2 представлен пример таблицы с кодировкой Типа характера.

| | Холерик | Флегматик | Меланхолик | Сангвиник |
|----------|---------|-----------|------------|-----------|
| Значение | 1.0.0.0 | 0.1.0.0 | 0.0.1.0 | 0.0.0.1 |
| | | | | |

Рис. 2. Таблица кодировки

Перед началом обучения необходимо нормализовать входные сигналы нейросети. Возраст можно представить значениями в диапазоне от 0 до 1. Например 18 лет – 0,18. Пол принимает значения 0 и 1.

На рис. 3 представлена модель многослойного перцептрона для классификации. В качестве модели классификации будем использовать многослойный перцептрон с одним

скрытым слоем. Во входном слое содержится пять нейронов – характеристики пользователей из анкеты. Скрытый слой состоит из 4 нейронов. В выходном слое три нейрона (жанры кинофильмов). Такая структура сети позволяет использовать для классификации нелинейную функцию активации. Принадлежность одному из трех жанров описывается бинарным вектором, принимающим значения (1,0,0), (0,1,0), (0,0,1).

Чтобы нейросеть научилась строить ассоциативные связи между входными характеристиками и выходными нейронами, определяющими класс пользователя, формируется обучающая выборка. Эта выборка включает данные по каждому зарегистрированному пользователю из заполненной им анкеты. Кроме того, в выборку по каждой записи включены фактические данные о выбранном пользователе фильме, соответствующем определенному жанру. На рис. 4 представлен вариант классификации пользователей в результате работы нейросети.

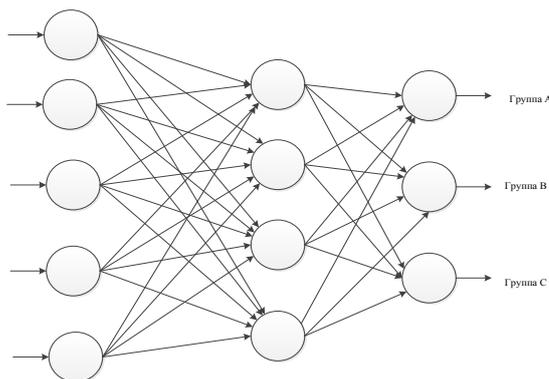


Рис. 3. Нейронная сеть для классификации

| | Группа А | Группа В | Группа С |
|----------------|----------|----------|----------|
| Пользователь 1 | | | |
| Пользователь 2 | | | |
| Пользователь 3 | | | |
| Пользователь 4 | | | |
| Пользователь 5 | | | |

Рис. 4. Результат классификации пользователей

Использование методов машинного обучения позволяют web-ориентированной информационной системе на основе встроенных процедур обучения автоматически находить в процессе работы наиболее эффективные стратегии, необходимые для правильной классификации, формирования групп пользователей с однотипными запросами, интересами и предпочтениями. На основе сформированных групп система может формировать релевантные рекомендации каждому пользователю из сформированной группы.

Библиографический список

1. Шполянская, И. Ю. Архитектура адаптивной Web-ориентированной системы взаимодействия с клиентами, основанной на технологиях Web Mining / И. Ю. Шполянская // Прикладная информатика. – 2017. – № 2 (68). – С. 5–12.
2. Bobadilla, J. Recommender systems survey / J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez // Knowledge-Based Systems. – 2013. – Vol. 46. – P. 109–132.
3. Adomavicius, G. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions / G. Adomavicius, A. Tuzhilin // IEEE Trans. Knowl. – 2005. – Data Eng., 17(6):734–749.
4. Adeniyi, D. A. Automated Web usage data mining and recommendation system using K-Nearest Neighbour (KNN) classification method / D. A. Adeniyi, Z. Wei, Y. Yongquan // Applied Computing and Informatics. – 2016. – Vol. 12, № 1. – P. 90–108.
5. Осовский, С. Нейронные сети для обработки информации / С. Осовский. – М. : Финансы и статистика, 2002. – 344 с.

Б. С. Ахметов, Л. М. Кыдыралина

Казахский национальный педагогический университет имени Абая,
Алматы, Казахстан

МОДЕЛЬ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ РЕШЕНИЯ ПО ФИНАНСОВЫМ ВЛОЖЕНИЯМ В КИБЕРБЕЗОПАСНОСТЬ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ УНИВЕРСИТЕТА

Предложена модель для оценки стратегий инвестирования в информационно-образовательную среду университета (ИОСУ). Одним из вариантов решения подобных задач и, в частности, оценивания рисков, связанных с финансированием систем защиты информации и кибербезопасности ИОСУ, является внедрение интеллектуализированных систем поддержки принятия решений. Подобные системы позволяют менеджменту учебных заведений принимать рациональные решения по вложению финансовых средств в развитие инструментария защиты ИОСУ. Предлагается модель для ИСППР по дискретной процедуре финансирования систем защиты информации и кибербезопасности ИОСУ.

Современные крупные учебные заведения (университеты, колледжи, студенческие кампусы и др.) и их корпоративные сети давно сформировали собственные информационно-образовательные среды (ИОС). Т.е. фактически ИОС – это многоуровневая иерархическая система. В подобных системах порой сталкиваются интересы и данные различных пользователей. В эпоху цифровизации и глобализации ИОС университетов, классическое понятие периметра корпоративной сети учебного заведения выглядит достаточно размытым. Сегодня трудно представить ИОС крупного университета без смартфонов, планшетов и иных устройств с веб-приложениями. Также продолжается внедрение технологий и устройств, предоставляющих возможности по получению доступа к учебным сервисам за пределами университета, например, из общежитий, библиотек, других учебных заведений и др.

Как отмечают многие специалисты по защите информации (ЗИ), системы информационной безопасности (ИБ) и кибербезопасности (КрБ) образовательных учреждений, в частности, университетов должны не только обеспечить сохранность информационных массивов и данных, в том числе конфиденциальных, но и гарантировать невозможность внешнего несанкционированного проникновения в информационно-коммуникационные системы учебных заведений (ИКСУ) [1–3]. Мероприятия по обеспечению ИБ и КрБ ИКСУ безусловно требуют финансовых вложений [4–6].

Отметим, что к защищаемым сведениям, которые хранятся и циркулируют в ИКСУ можно отнести [2, 4, 7]: персональные данные студентов (учащихся), преподавателей, сотрудников; оцифрованная информация, представляющая интеллектуальную собственность учебного заведения; информационные массивы, которые, обеспечивают учебный процесс, (например, мультимедийный контент, базы данных, обучающие программы); др.

Эта информация может выступить как объект хищения или искажения со стороны внешних (внутренних) компьютерных злоумышленников или из хулиганских побуждений, со стороны учащихся или сотрудников.

Все вышесказанное обуславливает актуальность проблематики исследований по разработке новых моделей, в частности для интеллектуализированных систем поддержки принятия решений в задачах выбора оптимальных финансовых стратегий стороны служб ИБ и КрБ ИКСУ.

Цель исследований – разработка модели для модуля интеллектуализированной системы поддержки принятия решений (ИСППР) в процессе выбора рациональных стратегий финансирования систем защиты и кибербезопасности ИКСУ и других учебных заведений.

Как показал анализ ряда последних работ [7–9] в данном сегменте исследований их общим недостатком представляется отсутствие обобщенных методик получения стратегий по финансированию в СЗИ и КрБ для ИКСУ. Кроме того, модели [8, 9], описывающие стратегии инвестирования в КрБ, не содержат упоминаний о ситуациях, когда имеется неопределенность в данных по финансовым стратегиям атакующей стороны. На наш взгляд, устранить недостатки многих имеющихся подходов в проблематике оптимизации стратегий финансирования СЗИ и КрБ ИКСУ можно путем применения методов теории дифференциальных и многошаговых игр качества [10].

В процессе исследований была предложена математическая модель для модуля системы поддержки принятия решений по финансированию средств кибербезопасности ИОС учебного заведения, и в частности, крупного университета с электронными системами и платформами для процесса обучения. Математическая модель базируется на теории многошаговых игр. Авторами, с учетом анализа предшествующих исследований в данной области, предложено решение, которое позволяет менеджменту информационной безопасности учебного заведения, производить предварительную оценку финансовых стратегий построения эффективных систем кибербезопасности информационно-образовательной среды. Отличительной чертой модели является допущение, что сторона защиты не имеет полной информации о финансовых стратегиях атакующей стороны и состояниях его финансовых ресурсов, направленных на преодоление рубежей кибербезопасности информационно-коммуникационных систем учебного заведения.

Безусловно обоим игрокам (защитнику ИОСУ (ЗИОСУ) и хакеру) требуются финансы и ресурсы (ФиР) для достижения своих целей. Атакующая сторона, например, может использовать стороннее коммерческое программное обеспечение (ПО) для взлома или подкупить персонал, обслуживающий ИОСУ. Будем полагать, что на заданный период времени $\{1, \dots, T\}$ (T – натуральное число) у ЗИОСУ выделено $de(0)$ ФиР, а у хакера – $ha(0)$. Эти параметры определяют прогнозную, в момент времени $t = 0$, величину ФиР, которыми обладают ЗИОСУ и хакер на достижение своих целей. В соответствии [10], принято, что для случая сопоставления решений двух игр – многошаговой и одношаговой, мы получим совпадение множеств начальных состояний ФиР ЗИОСУ и хакера со следующим свойством [2, 8].

Свойство 1: множество предпочтительности игрока, исходя из которых он достигает своей цели за T шагов, совпадает со множеством начальных состояний ФиР, из которых он достигает цели за один шаг при применении им оптимальной смешанной стратегии при оптимальном противодействии ему другим игроком в классе смешанных стратегий с вероятностью $1/T$.

В начальный момент времени t ЗИОСУ умножает величину финансового ресурса в момент времени 0 $de(0)$ на коэффициент (темп изменения, роста) $\alpha(t)$ и выбирает величину $u(t)$ ($u(t) \in [0, 1]$). Последняя позволит определить долю ФиР ЗИОСУ $\alpha(t) \cdot de(t)$, выделяемую им в момент времени t . Аналогично, в момент времени t хакер умножает величину $ha(t)$ на коэффициент (темп изменения, роста) $\beta(t)$ и выбирает величину $v(t)$ ($v(t) \in [0, 1]$). Это определит долю ФиР атакующей стороны (хакера) $\beta(t) \cdot ha(t)$, выделяемую им на взлом ИОСУ. Введем следующие обозначения: r_1 – коэффициент, показывающий, сколько ФиР потребуется хакеру, чтобы взломать ИОСУ, на защиту которого была израсходована единица ФиР ЗИОСУ; r_2 – коэффициент, показывающий, сколько ФиР потребуется ЗИОСУ, чтобы обезопасить системы ИОСУ, на взлом которых были израсходованы единица ФиР хакера.

Следовательно, с учетом [2, 8, 9], можно описать динамику изменения ФиР ЗИОСУ и атакующих такой системой дискретных уравнений:

$$de(t+1) = \alpha(t) \cdot de(t) - u(t) \cdot \alpha(t) \cdot de(t) - r_2 \cdot v(t) \cdot \beta(t) \cdot ha(t); \quad (1)$$

$$ha(t+1) = \beta(t) \cdot ha(t) - v(t) \cdot \beta(t) \cdot ha(t) - r_1 \cdot u(t) \cdot \alpha(t) \cdot de(t). \quad (2)$$

В момент времени t должно выполняться одно из условий:

1) $de(t) \geq 0, ha(t) < 0$. Если условие $1=true$, процедура финансирования системы кибербезопасности (СКБ) ИОСУ завершена. У атакующих не хватило ФиР преодолеть защиту ИОСУ;

2) $de(t) < 0, ha(t) \geq 0$. Если условие $2=true$, то процедура финансирования СКБ ИОСУ завершена. У ЗИОСУ не хватило ФиР для защиты информационно-коммуникационных систем образовательного учреждения.

3) $de(t) < 0, ha(t) < 0$. Если условие $3=true$, то процедура финансирования СКБ ИОСУ завершена. Ни у ЗИОСУ ни у хакера не хватило ФиР достичь своих целей.

4) $de(t) \geq 0, ha(t) \geq 0$. Если условие $4=true$, то процедура финансирования СКБ ИОСУ продолжится далее.

Значения $de(T), ha(T)$ иллюстрируют результаты выделения ФиР на СКБ ИОСУ. Выделение ФиР на СКБ ИОСУ в докладе рассмотрено в рамках схемы позиционной многошаговой игры с полной информацией [10].

Процесс выделения ФиР на СКБ ИОСУ рассматривается с двух точек зрения (задач):

1) Решение задачи с точки зрения первого игрока-союзника;

2) Решение задачи с точки зрения второго игрока-союзника [9, 10].

Поскольку задачи симметричны, в рамках тезисов рассматривается решение задачи с точки зрения первого игрока-союзника.

Обозначим через T^* множество $\{0, 1, \dots, T\}$.

Определение. Чистая стратегия первого игрока-союзника – это функция $u: T^* \cdot [0, 1] \cdot [0, 1] \rightarrow [0, 1]$, которая ставит состоянию информации (или позиции) $(t, (de, ha))$ значение $u(t, (de, ha)): 0 \leq u(t, (de, ha)) \leq 1$.

Следовательно, чистой стратегией первого игрока-союзника будет функция, которая ставит состоянию информации в момент времени t величину $u(t, (de, ha))$. Эта величина определяет долю ФиР ЗИОСУ, которую он планировал потратить на СКБ в момент t . Допускается, что игрок-противник (хакер) выбирает свое управляющее воздействие и размер своих ФиР, выделяемых на взлом ИОСУ, на основании любой информации. Определив стратегии в задаче 1, определим множество «предпочтительности» W_1 игрока ЗИОСУ. Тогда, W_1 – множество таких начальных состояний $(de(0), ha(0))$ ФиР ЗИОСУ и хакера, которые обладают ниже сформулированным свойством [10].

Свойство 2: для начальных состояний W_1 существует стратегия защитника, которая, для любых реализаций стратегии хакера, «приводит», в один из моментов времени t , состояние системы $(de(0), ha(0))$ в такое, при котором будет выполняться условие (1). При этом, у хакера отсутствует стратегия, которая может «привести» к выполнению условий (2) или (3), в один из предшествующих моментов времени.

Назовем *оптимальной* стратегией (финансовую составляющую) ЗИОСУ, обладающую *свойством 2*.

Решение задачи 1 заключается в нахождении множества предпочтительности ЗИОСУ. Также должны быть определены его оптимальные стратегии. Аналогично ставится задача с точки зрения хакера.

Решение задачи 1 находится с помощью инструментария теории многошаговых игр качества с полной информацией [2,3]. Данный инструментарий позволяет находить решение при любых соотношениях параметров игры. Аналогичным образом можно найти и множество предпочтительности для хакера. Полагаем при этом, что он обладает ограниченными финансовыми ресурсами. Таким же образом, решается *задача 2* с точки зрения

второго игрока-союзника. Это позволяет представить положительный ортант в плоскости $(de(0), ha(0))$ в виде трех множеств (конусов с вершиной в точке $(0,0)$). Одно множество (конус), примыкающий к оси OX , является множеством предпочтительным для ЗИОСУ. Второе множество (конус) является множеством предпочтительным для хакера. Третье множество (конус) является множеством нейтральным, с точки зрения обоих игроков.

В ходе исследований проведен вычислительный эксперимент. Решение игры приведено для всех случаев соотношения параметров игры для игроков со стороны защиты ИКСУ и компьютерных злоумышленников, стремящихся преодолеть рубежи кибербезопасности ИКСУ. Найдены варианты оптимального поведения (финансовой стратегии) защитника ИКСУ.

Библиографический список

1. Rezgui, Y. Information security awareness in higher education: An exploratory study / Y. Rezgui, M. Adam // *Computers & Security*. – 2010. – 27.7. – P. 241–253.
2. Sultan, N. Cloud computing for education: A new dawn? / N. Sultan // *International Journal of Information Management*. – 2010. – 30.2. – P. 109–116.
3. Ахметов, Б. С. Моделирование информационной образовательной среды вуза / Б. С. Ахметов, В. В. Яворский. – Караганда : КарГТУ, 2006. – С. 251.
4. Schneider, F. Cybersecurity education in universities / F. Schneider // *IEEE Security & Privacy*. – 2013. – 11.4. – P. 3–4.
5. Conklin, A. Cyber defense competitions and information security education: An active learning solution for a capstone course / A. Conklin // *System Sciences, 2006. HICSS'06: Proceedings of the 39th Annual Hawaii International Conference on*. – 2006. – Vol. 9. – URL: <http://computer.org>
6. Schuett, M. Information Security Synthesis in Online Universities / M. Schuett, M. Rahman // *arXiv preprint arXiv:1111.1771*. – 2011.
7. Radziwill, N. Cybersecurity Cost of Quality: Managing the Costs of Cybersecurity Risk Management / N. Radziwill, M. Benton. – URL: <https://arxiv.org/ftp/arxiv/papers/1707/1707.02653.pdf> - 2017
8. Development of the intelligent decision-making support system to manage cyber protection at the object of informatization / V. Lakhno, Y. Boiko, A. Mishchenko, V. Kozlovskii, O. Pupchenko // *Eastern-European Journal of Enterprise Technologies*. – 2017. – 2/9 (86). – P. 53–61.
9. Immune and defensive corporate systems with intellectual identification of threats / O. Petrov, B. Borowik, M. Karpinsky, O. Korchenko, V. Lakhno. – *Pszczyna : Śląska Oficyna Drukarska*, 2016. – P. 222.
10. Lakhno, V. Development of the decision making support system to control a procedure of financial investment / V. Lakhno, V. Malyukov, N. Gerasymchuk // *Eastern-European Journal of Enterprise Technologies*. – 2017. – Vol. 6, № 3. – P. 24–41.

Б. С. Башев, П. П. Макарычев

Пензенский государственный университет, Пенза, Россия

МЕТОД ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НИЗКОГО РАЗРЕШЕНИЯ В СИСТЕМАХ РАСПОЗНАВАНИЯ ТЕКСТА

Посвящена разработке высокопроизводительного метода распознавания текста при помощи OCR на зашумленных изображениях низкого разрешения. Представлен метод предобработки кадра изображения низкого разрешения с использованием бинарной пороговой обработки, бикубической интерполяции и морфологических фильтров для улучшения качества и количества распознанного текста.

Не смотря на стремительный прогресс в технологиях, технических процессах и программном обеспечении некоторые проблемы техники, если даже если и являются хорошо изученными, все равно проявляют себя в различных ситуациях.

К таким проблемам можно отнести – проблему зашумленности и низкого выходного разрешения изображения. И если на конечных устройствах пользователя, мы видим четкое изображение высоко разрешения, это не значит, что мы не получим такое же в попытке его продублировать или перенаправить [1]. Основные требования, предъявляемые к распознаваемому тексту это четкие границы текста, высокий контраст, правильная ориентация текста и наименьше возможное количество пиксельного шума.

Изображение, полученное с устройства захвата изображения, представляется как трехмерная матриц насыщенностей цветовых каналов BGR. Таким образом каждому цветовому пикселю изображения соответствует элемент матрицы. Элементы в матрице – принимают значения из множества $I \in \{0, 1, \dots, 255\}$. Для распознавания текста цветовая компонента изображения является избыточной. Преобразование цветного изображения происходит по формуле интенсивности $I = 0.299 R + 0.587 G + 0.144 B$. Изображение представляется как функция от координат пикселей. Таким образом, если к примеру, пиксель с координатами x и y белый, то мы можем записать это как $I(x, y) = 255$. Далее мы можем взять от нее производную от этой функции. Так как функция дискретная, то есть задана в ячейках, с установленным шагом между ними, то мы можем ее записать как (так как функция двумерная, то и производных, соответственно две):

$$\frac{\partial I(x, y)}{\partial x} = \frac{I(x + 1, y) - I(x - 1, y)}{2k} \quad (1)$$

$$\frac{\partial I(x, y)}{\partial y} = \frac{I(x, y + 1) - I(x, y - 1)}{2k} \quad (2)$$

В выражениях (1) и (2), $k = 1$ так как разница между соседними пикселями 1 пиксель. Таким образом, производные будут тоже матрицами такого же размера, как и исходное изображение. Имея производные по обеим координатам градиент:

$$\nabla I(x, y) = \left(\frac{\partial I(x, y)}{\partial x}, \frac{\partial I(x, y)}{\partial y} \right)$$

И его величину:

$$|\nabla I(x, y)| = \sqrt{\left(\frac{\partial I(x, y)}{\partial x} \right)^2 + \left(\frac{\partial I(x, y)}{\partial y} \right)^2}$$

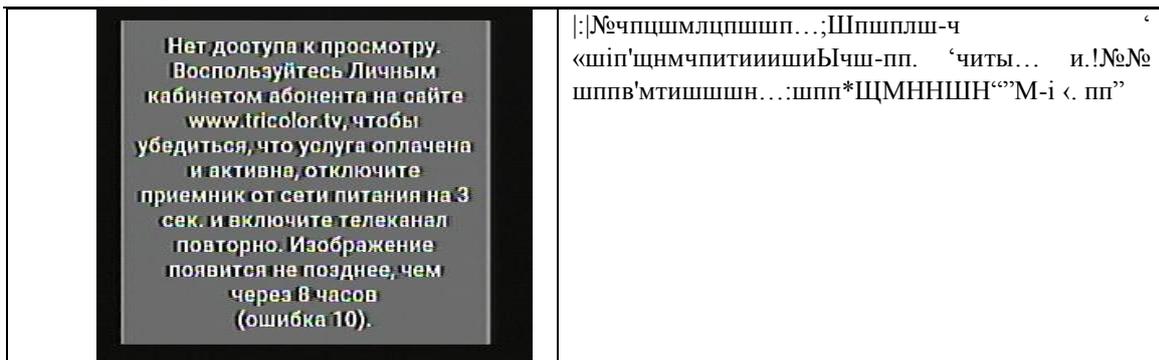
Величина градиента имеет большие значения там, где хранится нужная нам информация, т.е. текст. Увеличив точность оценки за счет избавления от шума, можно упростить работу алгоритму оптического распознавания текста (OCR) и повысить качество распознавания изображения [2].

Попытка распознавать текст напрямую передав тестовое изображение в функцию `pytesseract.image_to_string(frame, lang='RUS')` библиотеки Python-PyTesseract[7] без обработки, представлена в табл. 1.

Таблица 1

Распознавание текста оригинального изображения в 576p

| Изображение(ориг. размер) | Распознанный текст |
|---------------------------|--------------------|
|---------------------------|--------------------|



Существует несколько методов обработки изображений для увеличения градиента без существенной потери информации. Распространенным методом является метод подавления шумов [2, 3]. Существует несколько типов фильтров. Наибольшее распространение получили фильтры – медианно-гауссово размытие и билатеральная фильтрация [5]. Однако в условиях низкого разрешения применение этих фильтров, имеющих размерность 3x3 пикселя приводит к ухудшению распознавания. Вызвано это существенным уменьшением градиента по краям букв и цифр. Билатеральный фильтр в свою очередь хоть и повышает резкость изображения, плохо справляется с шумом сигнального характера. Подобная фильтрация не решает проблемы низкого разрешения изображения. Как видно из тестового распознавания, алгоритмы OCR чувствительны к размеру символов и регистру.

Предлагаемый в работе метод подразумевает предварительную обработку изображения при помощи бикубической интерполяции изображения до супер-разрешения (больше исходного в N раз) и сглаживания бинарного варианта, пороговой обработкой, результата, при помощи морфологических фильтров эрозии и дилатации [3].

Для решения поставленной задачи требуется искусственно увеличить размер обрабатываемого изображения. Тем самым мы не избавим изображения от шума, но создадим запас пикселей текста, потеря которых не будет влиять на его очертания и градиент. Для увеличения изображения в 5 раз, предлагается использовать метод бикубической интерполяции, программная реализация которого представлена в библиотеке Python – OpenCV [4,7]. Функция вызова имеет вид:

```
upscaled=cv2.resize(frame,None,fx=5,fy=5,interpolation=cv2.INTER_CUBIC).
```

Далее необходимо избавиться от всех градиентов фона, и компонент не относящиеся к тексту, сделав изображение бинарным. Это поможет исключить изменение цветовых компонент при конвертации и захвате. Вызовем функцию бинарной пороговой фильтрации (thresholding) [5].

```
ret, thresh1 = cv2.threshold(upscaled, thresh, 255, cv2.THRESH_BINARY).
```

Использования адаптивной пороговой фильтрации или метода Оцу[6], тоже возможно, но здесь стоит учитывать особенности предметной области. Параметр *thresh* необходимо подобрать экспериментально. В исследуемой предметной области используется белый текст на темном фоне, поэтому экспериментально найденный параметр *thresh* принимает значения *thresh* ∈ {120, ... 150}. После применения порогового фильтра изображения останутся только единичные шумы, вне текста. Можно было бы применить медианный фильтр, но так мы опять же потеряем контур символов. Гораздо эффективней будет применить подряд два морфологических фильтра – “эрозию” и “дилатацию” [5,7]. В результате обрабатываемое изображение избавится от шума вне текста, и станут более резкими очертания символов. Применение фильтров осуществляется с использованием операторов:

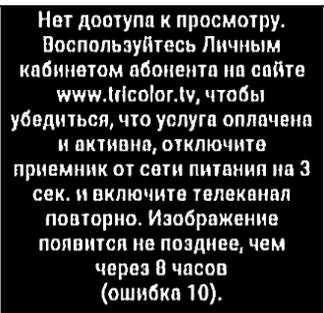
```
kernel = np.ones((1, 1), np.uint8);
img = cv2.dilate(img, kernel, iterations=1);
img = cv2.erode(img, kernel, iterations=1).
```

Предложенный метод использован для обработки тестового изображения. Результат распознавания текста в изображении осуществляется через библиотеку Python – PyTesseract [7] и приведен в табл. 2.

Из табл. 2, видно, что использование предложенного метода обработки изображения снижает уровень шума и избыточность на изображении, и позволяет обеспечить более высокую точность распознавание текста при низких разрешениях, ниже чем 720р. Недостатком предлагаемого метода является экспериментальный подбор параметра *thresh* бинарной пороговой фильтрации. Решение этой задачи во многом связано с ее предметной областью, относительно рассматриваемого изображения и выходит за рамки данной статьи.

Таблица 2

Распознавание текста обработанного изображения

| Изображение(миниатюра) | Примечание | Распознанный текст |
|---|--|--|
|  | <p>Английские символы/слова и переносы строк не учитываются</p> <p>Текст не попавший в порог(серый) threshold не распознался</p> | <p>Нет доступа к просмотру. Воспользуйтесь Личным кабинетом абонента на сайте www.tricolor.tv чтобы убедиться, что услуга оплачена и активна, отключите приемник от сети питания на 3 сек. и включите телеканал повторно.</p> <p>Изображение появится не позднее, чем через 8 часов(ошибка 10).</p> |

Библиографический список

1. Pratt, W. Digital image processing / W. Pratt. – N.Y : J. Wiley & Sons, 2001. – P. 735.
2. Mitzel, D. Video super resolution using duality based TV-L1 optical flow / D. Mitzel, T. Pock, T. Schoenemann, D. Cremers // DOGM. – 2009. – P. 432–441. – URL: <http://vision.in.tum.de>
3. Цифровая обработка изображений в информационных системах : учеб. пособие / И. С. Грузман, В. С. Киричук, В. П. Косых, Г. И. Перетягин, А. А. Спектор. – Новосибирск : НГТУ, 2002. – С. 84.
4. Farsiu, S. Fast and robust Super-Resolution / S. Farsiu, D. Robinson, M. Elad, P. Milanfa // Proc IEEE Int. Conf. on Image Process. – 2003. – P. 291–294. – URL: <http://recog.ru>
5. Gonzales, R. Digital image processing / R. Gonzales, R. Woods. – Prentice Hall, 2002. – P. 1070. – URL: <http://scirp.org>
6. Ping-Sung, Liao. A Fast Algorithm for Multilevel Thresholding / Liao Ping-Sung, Chen Tse-Sheng, Chung Pau-Choo // J. Inf. Sci.Eng. – 2001. – № 17. – P. 713–727.
7. Kaehler, A. Learning Opencv / A. Kaehler, G. Bradski. – M. : O'Reilly Media, 2008.– 580 p.

П. П. Макарычев

Пензенский государственный университет, Пенза, Россия

МОДЕЛЬ ДАТЧИКА РАССОГЛАСОВАНИЙ ДИРЕКЦИОННЫХ УГЛОВ В СЛЕДЯЩЕЙ СИСТЕМЕ

Обсуждается подход к построению модели датчика рассогласования дирекционных углов между заданным (начальным) направлением и направлением на объект наблюдения.

При определении вида функции рассогласования дирекционных углов двух направлений, как правило, приходится решать задачу интерполяции. Для решения этой задачи определяют значения функции $y = f(x)$ и производных $f'(x)$, $f''(x)$,... для нескольких зна-

чений аргументов $\{x_1, x_2, \dots, x_n\}$. По значениям аргумента, функции $\{y_1, y_2, \dots, y_n\}$ и производных функции задают более простую функцию интерполяции $y = F(x)$. Сплайновые модели по сравнению с другими моделями интерполяции отличаются повышенной интерполяционной точностью и хорошими аппроксимирующими свойствами. Наибольшее распространение получили кубические сплайны Эрмита, построенные на интервалах (x_k, x_{k+1}) с использованием кубических полиномов [1]. Для задания кубических полиномов автором предложены базисные функции, использование которых облегчает решение задачи построения локальных и натуральных сплайнов Эрмита [2, 3].

В данной работе рассматривается вариант построения модели датчика рассогласования с использованием локальных кубических эрмитовых сплайнов на основе предложенной системы базисных функций.

Предположим, что функция рассогласования дирекционных углов должна быть определена на интервале $(-180, +180)$. Разработчику датчика известны значения функций $U(\varphi)$, $U'(\varphi)$ при значениях аргумента $\varphi = 0$, $\varphi = +\varphi_0$, $\varphi = -\varphi_0$, $|\varphi_0| \leq 180$ град.

Для любого выделенного интервала угла рассогласования дирекционных углов можно задать функцию. Для интервала $0 \leq \varphi \leq \varphi_0$ функция $U(\varphi)$ при кубической интерполяции может иметь вид:

$$U(\varphi) = C_0 + C_1\varphi + C_2\varphi^2 + C_3\varphi^3, \quad (1)$$

где C_0, C_1, C_2, C_3 – коэффициенты, значения которых следует определить.

Для функции $U(\varphi)$ на интервале $0 \leq \varphi \leq \varphi_0$ должен выполняться набор условий. Например:

$$U(0) = 0, \quad U(\varphi_0) = U_0, \quad U'(0) = U_0/\varphi_0, \quad U'(\varphi_0) = 0. \quad (2)$$

Из выражений (2) следует, что функция и первая производная функции должны быть непрерывны на интервале $(-180, +180)$. На основе выражения (1) и заданных условий (2) можно составить систему уравнений следующего вида:

$$\begin{cases} C_0 = 0; \\ C_0 + C_1\varphi_0 + C_2\varphi_0^2 + C_3\varphi_0^3 = U_0; \\ C_1 = U_0/\varphi_0; \\ C_1 + 2C_2\varphi_0 + 3C_3\varphi_0^2 = 0. \end{cases} \quad (3)$$

Система уравнений (3) составлена в предположении, что условия (2) выполняются. Решение системы уравнений (3) позволяет определить значения всех коэффициентов функции $U(\varphi)$ на интервале $0 \leq \varphi \leq \varphi_0$:

$$C_0 = 0, \quad C_1 = U_0/\varphi_0, \quad C_2 = U_0/\varphi_0^2, \quad C_3 = -U_0/\varphi_0^3.$$

При найденных значениях коэффициентов функция датчика рассогласования $U(\varphi)$ на интервале $0 \leq \varphi \leq \varphi_0$ принимает вид:

$$U(\varphi) = \frac{U_0}{\varphi_0} \varphi + \frac{U_0}{\varphi_0^2} \varphi^2 - \frac{U_0}{\varphi_0^3} \varphi^3. \quad (3)$$

Аналогично решается задача на интервале значений аргумента $[-\varphi_0, 0]$. При этом функция $U(\varphi)$ имеет вид:

$$U(\varphi) = \frac{U_0}{\varphi_0} \varphi - \frac{U_0}{\varphi_0^2} \varphi^2 + \frac{U_0}{\varphi_0^3} \varphi^3. \quad (4)$$

Таким образом, если воспользоваться известным подходом к сплайновой кубической интерполяции, то необходимо вначале на интервалах, определяемых точками на оси $-180, -\varphi_0, 0, \varphi_0, 180$, построить 4 локальных сплайна, которые затем необходимо будет склеить [1].

Однако возможен и другой подход к построению локального сплайна. В соответствие с заданными условиями (2) для решения задачи зададим систему базисных функций следующего вида:

$$\begin{aligned} f_0(\varphi) &= 0; \\ f_1(\varphi) &= 1/2(|\varphi + \varphi_0| - |\varphi - \varphi_0|); \\ f_2(\varphi) &= 1/4(|\varphi + \varphi_0| - |\varphi - \varphi_0|) \cdot \||\varphi + \varphi_0| - |\varphi - \varphi_0|\|; \\ f_3(\varphi) &= 1/8(|\varphi + \varphi_0| - |\varphi - \varphi_0|)^3. \end{aligned} \quad (5)$$

На основе выражений (3, 4, 5) строим сплайн-функцию датчика рассогласования дирекционных углов в виде:

$$U(\varphi) = U_0/\varphi_0 f_1(\varphi) + U_0/\varphi_0^2 f_2(\varphi) + U_0/\varphi_0^3 f_3(\varphi).$$

Графики базисных функций $f_2(\varphi)$, $f_3(\varphi)$ и функции датчика рассогласования углов $U(\varphi)$ приведены на рис. 1.

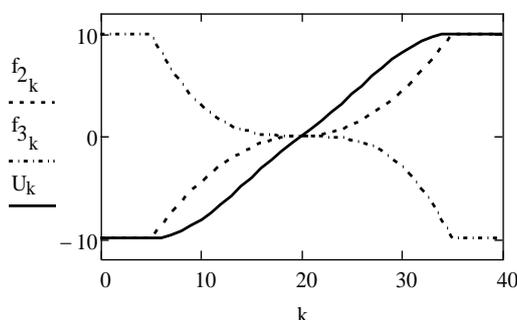


Рис. 1. Графики функций $f_2(\varphi)$, $f_3(\varphi)$ $U(\varphi)$

Графики функций на рисунке построены при следующих значениях $\varphi_0 = \pm 3$ град, $U(\varphi_0) = 10$ в, $U(-\varphi_0) = -10$ в, $\Delta\varphi = 0,2$ град. Базисные функции и функция рассогласования углов непрерывны и дифференцируемы. Параметры функции рассогласования могут корректироваться в зависимости от требований, предъявляемых к датчику рассогласования дирекционных углов разработчиком.

Выводы. Предложенный подход имеет достоинства. Во-первых, при разработке датчиков рассогласования дирекционных углов выполняется требование дифференцируемости функции интерполяции. Во-вторых, отпадает необходимость выполнения операций склеивания кубических полиномов при задании локального эрмитова сплайна.

Библиографический список

1. Алберг, Дж. Теория сплайнов и ее приложения / Дж. Алберг, Э. Нельсон, Дж. Уолш. – М. : Мир, 1972. – 316 с.
2. Макарычев, П. П. Моделирование многокомпонентных систем на основе маркированных графов : монография / П. П. Макарычев, М. А. Волгина. – Пенза : Изд-во ПГУ, 2011. – 155 с.
3. Макарычев, П. П. Модели функциональной зависимости в виде локального сплайна Эрмита / П. П. Макарычев // Актуальные вопросы современной науки: теория и практика научных исследований : сб. науч. тр. Всерос. науч.-практ. конф. – Пенза : Изд-во ПГТУ, 2017. – С. 49–52.

М. М. Алкезуини, В. И. Горбаченко

Пензенский государственный университет, Пенза, Россия

РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ НА СЕТЯХ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ

Разработана реализация метода Левенберга–Марквардта для обучения сетей радиальных базисных функций при решении краевых задач. Метод Левенберга–Марквардта при более простой реализации показал соизмеримые результаты по сравнению с методом доверительных областей.

Для численного решения краевых задач, описываемых дифференциальными уравнениями в частных производных (ДУЧП), получают популярность бессеточные методы [1], не требующие сложной процедуры построения сетки. Среди бессеточных методов особой популярностью пользуются проекционные методы, использующие в качестве базисных функций радиальные базисные функции (РБ-функции) [2]. Методы, использующие РБ-функции, позволяют получить дифференцируемое решение в любой точке области решения, применимы к задачам произвольной размерности со сложными расчетными областями. При решении задачи параметры базисных функций выбираются, а веса вычисляются так, чтобы невязки, получаемые после подстановки приближенного решения в пробных точках в уравнение, были равны нулю. Недостаток таких методов – необходимость неформального подбора параметров базисных функций. Этот недостаток преодолевается применением специального класса нейронных сетей – сетей радиальных базисных функций (РБФ-сетей) [3–4]. Применение РБФ-сетей позволяет в процессе обучения сетей настраивать как веса, так и параметры РБ-функций.

Решение задачи формируется в процессе обучения РБФ-сети. Поэтому актуальной является разработка быстрых алгоритмов решения краевых задач на РБФ-сетях. Целью данной работы является адаптация современных ускоренных градиентных методов первого порядка и метода Левенберга–Марквардта для обучения РБФ-сетей при решении краевых задач. Данная работа является развитием работы авторов [8], в которой перечисленные методы применены для решения задач аппроксимации функций с помощью РБФ-сетей.

РБФ-сеть включает два слоя [6]. Первый слой состоит из РБ-функций, производящих нелинейное преобразование входного вектора $\mathbf{x} = [x_1, x_2, \dots, x_d]$ – координат точки, в которой вычисляется приближение к решению (d – размерность пространства). РБ-функция – это функции расстояния точки пространства от параметра функции, называемого центром функции: $\varphi(\|\mathbf{x} - \mathbf{c}\|, \mathbf{p})$, где \mathbf{x} – точка пространства, \mathbf{c} – центр радиальной базисной функции, $\|\mathbf{x} - \mathbf{c}\|$ – евклидова норма (расстояние) между точкой и центром, \mathbf{p} – вектор параметров функции. Применяются различные РБ-функции. В данной работе используется функция Гаусса (Гауссиан)

$\varphi(\|\mathbf{x} - \mathbf{c}\|, a) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}\|^2}{2a^2}\right)$, где \mathbf{c} – положение центра функции, a – параметр формы,

часто называемый шириной. Второй слой РБФ-сети представляет собой линейный взвешенный сумматор, при решении краевых задач описываемый выражением

$$u(\mathbf{x}) = \sum_{m=1}^M w_m \varphi_m(\mathbf{x}; \mathbf{p}_m), \quad (1)$$

где M – количество РБ-функций, w_m – вес РБ-функции φ_m , \mathbf{p}_m – вектор параметров РБ-функции φ_m .

Решение краевых задач на РБФ-сетях рассмотрим на примере решения стационарной краевой задачи, описывающей процесс в однородной среде

$$Lu(\mathbf{x}) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad Bu(\mathbf{x}) = p(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \quad (2)$$

где u – искомое решение; L – дифференциальный оператор, например, $L = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$; B – оператор граничных условий; Ω – область решения; $\partial\Omega$ – граница области; f и p – известные функции.

Обучение сети, т.е. настройка весов \mathbf{w} и параметров РБ-функций \mathbf{p} , основано на минимизации функционала ошибки, представляющего собой сумму квадратов невязок в пробных точках для уравнения и граничных условий

$$I(\mathbf{w}, \mathbf{p}) = \sum_{i=1}^N [Lu_{\text{RBF}}(\mathbf{x}_i; \mathbf{w}, \mathbf{p}) - f(\mathbf{x}_i)]^2 + \lambda \sum_{i=N+1}^{N+K} [Bu_{\text{RBF}}(\mathbf{x}_i; \mathbf{w}, \mathbf{p}) - p(\mathbf{x}_i)]^2 \rightarrow \min, \quad ()$$

где $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \in \Omega$, $\mathbf{x}_{N+1}, \mathbf{x}_{N+2}, \dots, \mathbf{x}_{N+K} \in \partial\Omega$; λ – подбираемый штрафной множитель, учитывающий вклад невязок в граничных пробных точках, u_{RBF} – приближение к решению (), полученное РБФ-сетью, N, K – количество пробных точек внутри и на границе области решения соответственно.

Для обучения РБФ-сетей применяются градиентные методы. Среди градиентных методов выделяют методы нулевого порядка, использующие только информацию о значении минимизируемой функции, методы первого порядка, использующие первые производные минимизируемой функции, т.е. градиент функции, и методы второго порядка, использующие вторые производные – матрицу Гессе. В известных работах [3] используется метод первого порядка – метод градиентного спуска. Настройка вектора $\boldsymbol{\theta}$ параметров РБФ-сетей на k -й итерации градиентного спуска описывается следующим образом

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \Delta\boldsymbol{\theta}^{(k+1)}, \quad ()$$

где $\Delta\boldsymbol{\theta}^{(k+1)} = -\eta \nabla I(\boldsymbol{\theta}^{(k)})$ – поправка вектора $\boldsymbol{\theta}$, η – подбираемый числовой коэффициент (скорость обучения), $\nabla I(\boldsymbol{\theta}^{(k)})$ – вектор градиента функционала () по значению параметра $\boldsymbol{\theta}^{(k)}$ на итерации k .

Недостатком метода градиентного спуска является низкая скорость сходимости, что не позволяет получить высокую точность решения.

Методы второго порядка не получили распространения при обучении РБФ-сетей. Например, популярный метод Левенберга-Марквардта практически не применяется для обучения РБФ-сетей. Известны только отдельные примеры применения этого метода [7–9] в областях, не связанных с решением краевых задач. Для обучения РБФ-сетей перспективным является метод доверительных областей, обладающий высокой скоростью сходимости. Использование метода доверительных областей для обучения РБФ-сетей при решении краевых задач предложено в [10] и развито в [11–12]. Но этот метод достаточно сложен, так как на каждой итерации обучения сети приходится решать задачу условной минимизации.

Рассмотрим применение современных методов первого порядка и метода Левенберга-Марквардта для обучения РБФ-сетей. Метод градиентного спуска с импульсом [13] отличается от метода градиентного спуска другим вычислением вектора поправки в формуле коррекции вектора параметров сети ():

$$\Delta\boldsymbol{\theta}^{(k+1)} = \alpha \Delta\boldsymbol{\theta}^{(k)} - \eta - \eta \nabla I(\boldsymbol{\theta}^{(k)}), \quad ()$$

где $\nabla I(\boldsymbol{\theta}^{(k)})$ – градиент функционала (), η – скорость обучения, α – коэффициент момента, принимающий значения в интервале [0, 1].

Слагаемое в (), не зависящее от градиента, оказывает наибольшее влияние на изменение вектора поправки в областях с малым значением градиента. Можно сказать, метод позволяет «по инерции» преодолевать такие области на плоских участках функционала ошибки и вблизи локальных максимумов.

При обучении нейронных сетей глубокой архитектуры популярен метод ускоренного градиента Нестерова (NAG – NesterovAcceleratedGradient) [13]. В методе NAG поправка вектора параметров сети описывается следующим образом

$$\Delta\theta^{(k+1)} = \alpha\Delta\theta^{(k)} - \eta\nabla I\left(\theta^{(k)} + \alpha\Delta\theta^{(k)}\right).$$

Метод NAG отличается от градиентного спуска с импульсом вычислением градиента в точке, учитывающей коррекцию параметров.

В реализации методов первого порядка компоненты вектора градиента функционала ошибки по весам, центрам и ширине несложно вычислить аналитически. В качестве примера рассмотрим модельную задачу, описываемую уравнением Лапласа с граничными условиями Дирихле

$$\frac{\partial^2 u}{\partial x_1^2} + \frac{\partial^2 u}{\partial x_2^2} = f(x_1, x_2), \quad (x_1, x_2) \in \Omega, \quad u = p(x_1, x_2), \quad (x_1, x_2) \in \partial\Omega, \quad ()$$

где $\partial\Omega$ – граница области; f и p – известные функции (x, y) .

Тогда функционал ошибки () запишется в виде

$$I = \left[\sum_{i=1}^N (\Delta u_i - f_i)^2 + \lambda \cdot \sum_{j=1}^K (u_j - p_j)^2 \right], \quad ()$$

где Δu_i – Лапласиан в точке i .

Компоненты градиента по весам РБФ-сети вычисляются по формуле

$$\frac{\partial I}{\partial w_p} = \sum_{i=1}^N (\Delta v_i - f_i) \cdot e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2}{2a_p^2}} \cdot \frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2 - 2a_p^2}{a_p^4} + \lambda \sum_{j=1}^K (v_j - p_j) w_p e^{-\frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{2a_p^2}}.$$

Компонент градиента по координате c_{p1} центра РБ-функций имеет вид

$$\frac{\partial I}{\partial c_{p1}} = w_p \sum_{i=1}^N (\Delta v_i - f_i) e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2}{2a_p^2}} (x_{i1} - c_{p1}) \frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2 - 4a_p^2}{a_p^6} + \lambda w_p \sum_{j=1}^K (v_j - p_j) e^{-\frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{2a_p^2}} \frac{(x_{j1} - c_{p1})}{a_p^2}$$

Аналогичный вид имеет компонент градиента по координате c_{p2} .

Компонент градиента по ширине равен

$$\begin{aligned} \frac{\partial I}{\partial a_p} = w_p \sum_{i=1}^N (\Delta v_i - f_i) \frac{\|\mathbf{x}_i - \mathbf{c}_p\|^4 - 6a_p \|\mathbf{x}_i - \mathbf{c}_p\|^2 + 4a_p^4}{a_p^7} e^{-\frac{\|\mathbf{x}_i - \mathbf{c}_p\|^2}{2a_p^2}} \\ + \lambda w_p \sum_{j=1}^K (v_j - p_j) \frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{a_p^3} e^{-\frac{\|\mathbf{x}_j - \mathbf{c}_p\|^2}{2a_p^2}}. \end{aligned}$$

Процесс обучения сети заканчивается при малом значении функционала ошибки () или средней квадратической погрешности.

Рассмотрим адаптацию для обучения РБФ-сети метода Левенберга-Марквардта на примере задачи (). Введем вектор параметров сети

$$\theta = \left[w_1, w_2, \dots, w_M, c_{11}, c_{21}, \dots, c_{M,1}, c_{12}, c_{22}, \dots, c_{M,2}, a_1, a_2, \dots, a_M \right]^T,$$

где параметры j -й РБ-функции ($j = 1, 2, 3, \dots, M$): w_j – вес, c_{j1} и c_{j2} – координаты центра, a_j – ширина.

В методе Левенберга-Марквардта поправка $\Delta\theta^{(k)}$ вектора параметров θ находится из решения системы линейных алгебраических уравнений

$$(\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}) \Delta\theta^{(k)} = -\mathbf{g}_{k-1}, \quad ()$$

где матрица $\mathbf{J}_{k-1}^T \mathbf{J}_{k-1} + \mu_k \mathbf{E}$ – аппроксимация матрицы Гессе, \mathbf{E} – единичная матрица, μ_k – параметр регуляризации, изменяющийся на каждом шаге обучения, $\mathbf{g} = \mathbf{J}^T \mathbf{e}$ – вектор градиента функционала () по вектору параметров θ , $\mathbf{e} = [e_1 \ e_2 \ \dots \ e_n]^T$ – вектор невязок во внутренних и граничных пробных точках, \mathbf{J}_{k-1} – матрица Якоби, вычисленная в $k-1$ итерации.

Представим матрицу Якоби в блочном виде $\mathbf{J} = [\mathbf{J}_w \mid \mathbf{J}_{c_1} \mid \mathbf{J}_{c_2} \mid \mathbf{J}_a]$. Элементы матрицы Якоби несложно вычислить аналитически. Элементы матрицы \mathbf{J}_w для внутренних пробных точек вычисляются по формуле

$$\frac{\partial e_i}{\partial w_j} = \frac{\partial(\Delta v_i - f_i)}{\partial w_j} = e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2a_j^2}} \cdot \frac{\|\mathbf{x}-\mathbf{c}_j\|^2 - 2a_j^2}{a_j^4}$$

Для граничных пробных точек вычисления производятся по формуле $\frac{\partial e_i}{\partial w_j} = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2a_j^2}\right)$. Элементы матрицы \mathbf{J}_{c_1} для внутренних пробных точек имеют вид

$$\frac{\partial e_i}{\partial c_{j1}} = \frac{w_j}{a_j^4} \cdot e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2a_j^2}} \cdot (x_1 - c_{j1}) \cdot \frac{\|\mathbf{x}-\mathbf{c}_j\|^2 - 4a_j^2}{a_j^2}$$

Для граничных точек элементы матрицы записываются в виде

$$\frac{\partial e_i}{\partial c_{j1}} = w_j \cdot e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2a_j^2}} \cdot \frac{(x_1 - c_{j1})}{a_j^2}$$

Аналогично вычисляются элементы матрицы \mathbf{J}_{c_2} .

Элементы матрицы \mathbf{J}_a для внутренних пробных точек вычисляются по формуле

$$\frac{\partial e_i}{\partial a_j} = \frac{w_j}{a_j^5} \cdot e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2a_j^2}} \cdot \left[\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{a_j^2} \cdot (\|\mathbf{x}-\mathbf{c}_j\|^2 - 2a_j^2) - 4 \cdot (\|\mathbf{x}-\mathbf{c}_j\|^2 - a_j^2) \right]$$

Для граничных точек элементы матрицы записываются в виде

$$\frac{\partial e_i}{\partial a_j} = w_j \cdot e^{-\frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{2a_j^2}} \cdot \frac{\|\mathbf{x}-\mathbf{c}_j\|^2}{a_j^3}$$

Условия завершения процесса обучения по методу Левенберга-Марквардта те же, что и в методе Нестерова.

В методе Левенберга-Марквардта параметр регуляризации μ должен изменяться в процессе обучения сети. Процесс обучения начинается при относительно большом значении параметра μ . Это означает, что в начале процесса обучения Гессиан в () близок к приближенному значению $\mathbf{H} \approx \mu \mathbf{E}$, а вектор поправки определяется методом градиентного спуска с малым шагом $\Delta\theta^{(k)} = -\mathbf{g}_{k-1}/\mu_k$. По мере уменьшения функционала ошибки параметр μ уменьшается и метод приближается к методу Ньютона с аппроксимацией Гессиана

$\mathbf{H} \approx \mathbf{J}^T \mathbf{J}$. Это обеспечивает высокую скорость сходимости, так как метод Ньютона вблизи минимума функционала ошибки имеет хорошую сходимость.

В [14] показано, что метод Левенберга-Марквардта эквивалентен методу доверительных областей, а радиус доверительной области регулируется параметром μ . Но в отличие от известных реализаций метода доверительных областей метод Левенберга-Марквардта не требует решения на каждой итерации обучения достаточно сложной задачи условной оптимизации. То есть метод Левенберга-Марквардта сохраняя положительные свойства метода доверительных областей, является более простым.

Экспериментальное исследование рассмотренных методов проводилось на примере решения модельной задачи () при следующих функциях правой части и граничных условий $f(x_1, x_2) = \sin(\pi x_1) \cdot \sin(\pi x_2)$, $p(x_1, x_2) = 0$. Задача решалась в единичном квадрате. Количество внутренних пробных точек равно $N = 100$. Количество граничных пробных точек равно $K = 40$. Штрафной коэффициент функционала ошибки по границе равен $\lambda = 10$. Центры РБ-функций располагались регулярно на квадратной сетке с количеством центров по каждой координате, равным 8. Как показали эксперименты, целесообразно расположить центры и на расстоянии одного шага сетки за пределами области решения. Пробные точки располагались случайным образом в области решения и на границе области. Веса иницировались нулевыми значениями. Начальная ширина всех РБ-функций была постоянной, равной 0,2 для всех методов. Итерационный процесс обучения продолжался до достижения заданного значения средней квадратической ошибки. Зависимость среднеквадратической ошибки различных методов от номера итерации показана на рис. 1.

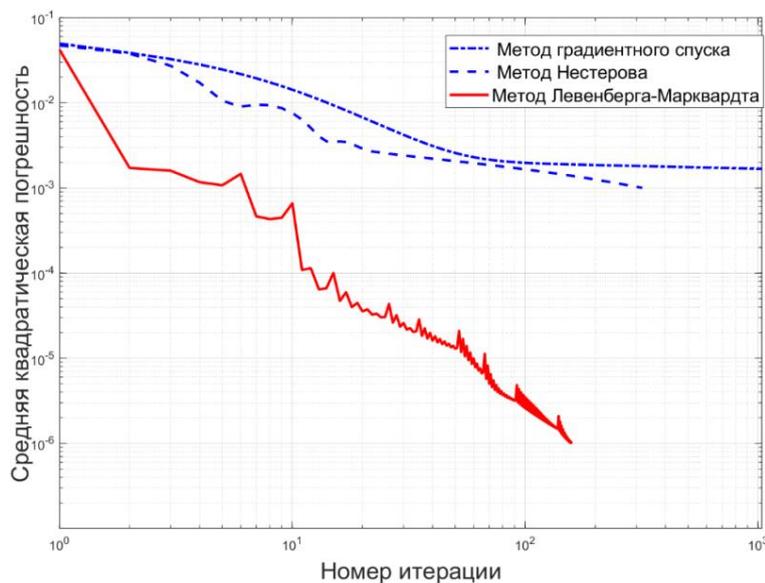


Рис. 1. Зависимость среднеквадратической ошибки от номера итерации

Эксперименты показали качественное преимущество метода Левенберга-Марквардта перед методами первого порядка. Метод градиентного спуска позволил решить модельную задачу с небольшой точностью. Для решения с большой точностью метод практически неприменим. Несколько бóльшую точность обеспечивает метод Нестерова. Только метод Левенберга-Марквардта позволил решить задачу с высокой точностью за приемлемое время. Метод Левенберга-Марквардта показал соизмеримые результаты по сравнению с методом доверительных областей [4], но реализация метода Левенберга-Марквардта проще.

Библиографический список

1. Meshfree Methods for Partial Differential Equations / ed. M. Griebel, Marc. A. Schweitzer. – Berlin : Springer, 2008. – 412 p.

2. Chen, W. Recent Advances in Radial Basis Function Collocation Methods / W. Chen, Z.-J. Fu Recent. – Berlin : Springer, 2014. – 90 p.
3. Yadav, N. An Introduction to Neural Network Methods for Differential Equations / N. Yadav, A. Yadav, M. Kumar. – Berlin : Springer, 2015. – 115 p.
4. Горбаченко, В. И. Решение краевых задач математической физики с помощью сетей радиальных базисных функций / В. И. Горбаченко, М. В. Жуков // Журнал вычислительной математики и математической физики. – 2017. – Т. 57, № 1. – С. 115–126.
5. Алкезуини, М. М. Совершенствование алгоритмов обучения сетей радиальных базисных функций для решения задач аппроксимации / М. М. Алкезуини, В. И. Горбаченко // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 3 (23). – С. 123–138.
6. Хайкин, С. Нейронные сети: полный курс / С. Хайкин. – М. : Вильямс, 2006. – 1104 с.
7. Zhang, L. A New Discrete-Continuous Algorithm for Radial Basis Function Networks Construction / L. Zhang, K. Li, H. He, G.W. Irwin // IEEE Trans. Neural Networks and Learning Systems. – 2013. – Vol. 24, № 11. – P. 1785–1798.
8. Fast and Efficient Second-Order Method for Training Radial Basis Function Networks / T. Xie, H. Yu, J. Hewlett, P. Rozycki, B. Wilamowski // IEEE Trans. Neural Networks and Learning Systems. – 2012. – Vol. 23, № 4. – P. 609–619.
9. An, R. An improved Levenberg – Marquardt algorithm with adaptive learning rate for RBF neural network / R. An, W. J. Li, H. G. Han, J. F. Qiao // Proceedings 2016 of the 35th Chinese Control Conference. – P. 3630–3635. – URL: <http://semanticscholar.org>
10. Горбаченко, В. И. Обучение сетей радиальных базисных функций методом доверительных областей для решения уравнения Пуассона / В. И. Горбаченко, М. В. Жуков // Информационные технологии. – 2013. – № 9. – С. 65–70.
11. Alqezweeni, M. M. Efficient Solving of Boundary Value Problems Using Radial Basis Function Networks Learned by Trust Region Method / M. M. Alqezweeni, V. I. Gorbachenko, M. V. Zhukov, M. S. Jaafar // International Journal of Mathematics and Mathematical Sciences. – 2018. – Vol. 2018. – P. 4.
12. Елисов, Л. Н. Обучение методом доверительных областей сетей радиальных базисных функций при решении краевых задач / Л. Н. Елисов, В. И. Горбаченко, М. В. Жуков // Автоматика и телемеханика. – 2018. – № 9. – С. 95–105.
13. Гудфеллоу, Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – М. : ДМК Пресс, 2018. – 652 с.
14. Marquardt, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters / D. W. Marquardt // Journal of the Society for Industrial and Applied Mathematics. – 1963. – Vol. 11, № 2. – P. 431–441.

К. Н. Савинов, Д. А. Данилин

Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ ПАКЕТОВ ПЛАТФОРМЫ .NETFRAMEWORK ДЛЯ СОЗДАНИЯ СЕТИ НЕЙРОНОВ «КРАМЕРА–ФОН МИЗЕСА»

Показано, что математические пакеты платформы .NetFramework, а также сторонние пакеты, разработанные на их основе, позволяют создавать искусственные нейронные сети любой сложности, в частности, состоящие из нейронов «Крамера–фон Мизеса», что свидетельствует о перспективности использования данной платформы в области изучения эффективности данных нейронов при решении различных задач.

В настоящее время разработка различного программного обеспечения, работа которого построена на использовании нейронных сетей, осуществляется, в общем случае, двумя способами.

Первый из них – использование специальных библиотек, пакетов и сборок, предоставляющих готовые решения для создания нейронных сетей определенного типа, например – фреймворка Keras для языка программирования Python[N]. Данный способ удобен для разработчика тем, что он требует от него меньших знаний математических основ работы нейронных сетей, а необходимость построения архитектуры сети и структуры нейрона если не исключена вовсе, то требует гораздо меньше времени, нежели при их «ручном» определении.

Однако данный способ неприменим в тех случаях, когда разрабатываемое ПО предполагает использование нестандартных типов нейронов – таких, как искусственные нейроны «Крамера-фон Мизеса», получивших меньшее распространение, чем, например, «Сигмоидальные» нейроны [3]. В этой ситуации разработчику приходится выбирать второй из способов, описанных выше – использование математических пакетов целевого языка программирования.

Декомпозиция работы сети нейронов «Крамера-фон Мизеса» на отдельные математические вычисления показывает, что в процессе обучения и проверки работы нейросети над входными данными (образами и их составляющими) выполняются описанные ниже операции.

Получение псевдослучайных целых чисел для генерации таблицы связей нейронов поводится в соответствии с требованиями о случайности подключения нейронов ко входам сети из ГОСТ Р 52633.5 [1]. При этом тип распределения вероятности для данного генератора явно не указан. В платформе .NetFramework(язык программирования C# 7.0) данное действие позволяет выполнить специальный класс генератора псевдослучайных чисел Random [2].

Операции возведения в степень, извлечения квадратного корня и получения модуля числа выполняются как в процессе обучения, так и в процессе проверки работы нейросети для вычисления вектора математических ожиданий – $\bar{E}(v_i)$, вектора стандартных отклонений – $\bar{\sigma}(v_i)$, средних значений вероятности для упорядоченной выборки биометрических параметров образа «Свой» $E(P(\tilde{v}_1)), E(P(\tilde{v}_2)), \dots, E(P(\tilde{v}_n))$. В платформе .NetFramework эти операции представлены методами класса Math -Pow, Sqrt, Abs соответственно.

Ключевым элементом является вычисление функции вероятности для сравнения представленных нейрону данных (образы «Свой» или «Чужой») с эталоном (получен на базе выборки образов «Свой»). Данная задача является наиболее сложной из всех необходимых операций, так как стандартный класс Math и иные классы, поставляемые компанией Microsoft пакетами .NetFramework, не имеют методов вычисления функции вероятности для нормального распределения вероятности (Гаусса). Для решения этой задачи разработчику ПО необходимо написать свой собственный метод вычисления функции $f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$ или использовать сторонний математический пакет Alglib, включающий в себя библиотеку для работы в области статистики, комбинаторики и теории вероятности [4]. Операция вычисления функции вероятности в данном пакете представлена методом Normaldistribution.

Операции сложения, вычитания, умножения, сравнения – используются на всех этапах работы с сетью нейронов и представлены базовой библиотекой языка программирования C# 7.0.

Таким образом, можно сделать вывод о том, что математические пакеты платформы .NetFramework обладают всеми методами и классами, необходимыми для создания нейронных сетей, а также позволяют разработчикам без особых усилий реализовывать модели различных искусственных нейронов, гибко меняя при этом их внутреннюю структуру. Все это позволяет говорить о перспективности платформы .NetFramework в области создания и изучения новых типов нейросетей.

Библиографический список

1. ГОСТ Р 52633.5–2011. Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей «биометрия–код доступа».
2. Троелсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троелсен, Ф. Джепикс. – М. : Вильямс, 2018 – 1328 с.

3. Иванов, А. И. Подавление шумов квантования биометрических данных при использовании многомерного критерия Крамера–фон Мизеса / А. И. Иванов, А. И. Газин, С. Е. Вятчанин // Проблемы информационной безопасности. Компьютерные системы. – 2016. – № 2. – С. 21–28.
4. Рашид, Т. Создаем нейронную сеть / Т. Рашид. – М. : Вильямс, 2018. – 272 с.

Д. В. Спирин

Пензенский государственный университет, Пенза, Россия

АЛГОРИТМ АВТОМАТИЗИРОВАННОГО ОБУЧЕНИЯ ДИАЛОГОВЫХ СИСТЕМ

При разработке диалоговых система одной из главной проблем является сложность их обучения, также чаще всего после обучения система представляет собой «черный ящик», изменение которого проблематично. Разработаны алгоритмы, направленные на решение данных проблем.

Введение

Виртуальный собеседник чаще всего представляет из себя автомат, в одном из состояний которого в текущий момент времени находится собеседник. Контекстом подобного диалога является как история общения, так и все состояния, через которые прошел собеседник в результате общения с виртуальной системой. Каждое состояние автомата – фрейм, полно описывающий текущее состояние, а также возможные переходы. Это системы с закрытой областью, ориентированные на решение задачи, так как фреймы описывают полно текущее состояние, и невозможно описать все возможные ситуации диалога. Рассматриваемые алгоритмы обучения достраивают автомат, по которому шло общение в диалоге.

Алгоритм обучения по истории диалогов

Если диалоговая система заменяет оператора, то она может быть обучена на истории диалогов. Здесь алгоритм выявляет деревья в диалогах и создает эквивалентный автомат. Во время работы алгоритма происходит попытка минимизировать количество деревьев и состояний. Происходит поиск по уже имеющимся состояниям. Несколько подряд идущих сообщений от оператора или собеседника соединяются в одно, если между ними не прошло достаточно времени (промежуток времени задается отдельно).

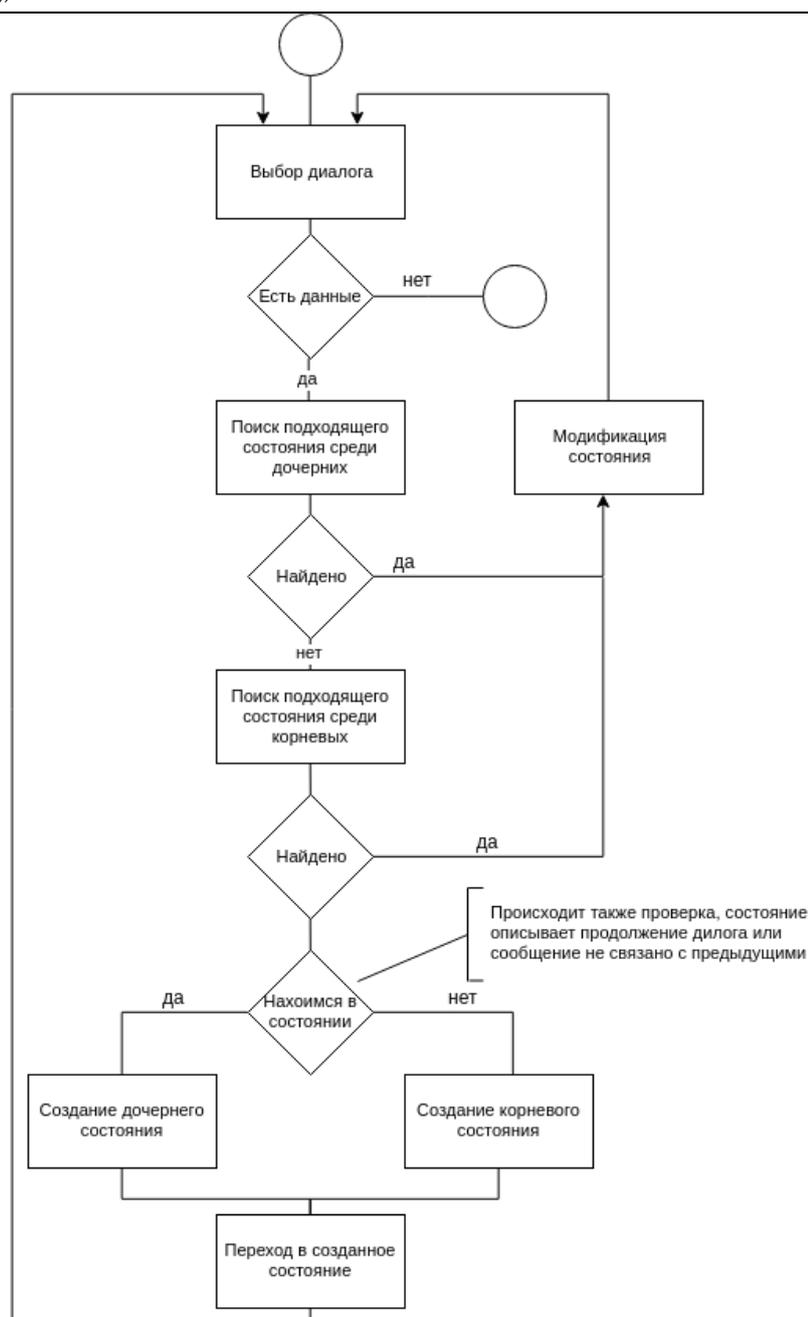


Рис. 1. Схема алгоритма обучения по истории диалогов

Так как обрабатывается вся история, первое сообщение всегда – начальное. Сначала проводится поиск корня, удовлетворяющего условию. Если корень найден, то следующее сообщение будет проверяться только среди узлов, дочерних тому, в которое они вошли, иначе – создается новое дерево и проверок больше не происходит.

Из схемы видно, что алгоритм проверяет, существует ли подходящее состояние в автомате для реплики, и если оно отсутствует, то алгоритм создает его. В случае, если подходящее состояние уже существует, то происходит его модификация с учетом информации, полученной из текущей реплики.

Алгоритм выполняется раз за разом для каждого нового диалога и для каждой реплики в нем. В результате работы алгоритма должен получиться лес из деревьев, каждое дерево которого разветвляется на возможные варианты развития диалогов.

Алгоритм обучения в режиме реального времени

Так как диалоговые системы в текущем состоянии не способны полностью заменить человека, алгоритм предполагает совместную работу диалоговой системы и оператора.

В момент, когда бот не знает ответа на реплику пользователя, в диалог включается оператор. Диалоговая система анализирует реплику пользователя и ответ на нее оператора. На основе этих данных создается дополнительное состояние в автомате. Для того, чтобы не потерять контекст диалога, состояние связано с предыдущим и не является начальным. Если реплика является первой в диалоге – будет создано начальное состояние автомата. После создания нового состояния, диалог переходит в это состояние.

Так как многие вопросы не связаны с контекстом, то созданные состояния оператор в дальнейшем может пометить начальными и отредактировать фрейм, созданный в процессе обучения.

Ниже представлена схема алгоритма.

Из схемы видно, что обучение работает только в случае, когда включается в диалог оператор. В остальное время обучения не происходит.

Результат обучения

Так как в работе алгоритма присутствует перебор всех состояний, в которые может быть осуществлен переход, то при увеличении размера обучающей выборки, время обучения на каждом последующем диалоге возрастало. Сложность алгоритма стремится к $O(n)$, где n количество начальных состояний автомата. В дальнейшем необходимо усовершенствовать реализацию, применив хеширование к сортировке состояний диалога для ускорения поиска подходящего перехода.

Заключение

Таким образом, на основании проведенного анализа предлагается использовать данные алгоритмы совместно. На первом этапе создания диалоговой системы необходимо проводить ее обучение на истории диалогов, а затем перманентно обучать в режиме реального времени.

Описанные алгоритмы обучения являются автоматизированными, а не автоматическими по нескольким причинам, описанным ниже.

Алгоритм не способен определить какое из создаваемых состояний может быть начальным, так как он не анализирует контекст. За начальные состояния принимаются только те, с которых начинался диалог. Если количество диалогов значительно, а их глубина невелика, то алгоритм показывает лучшие результаты обучения. После обучения необходимо в ручном режиме выбрать состояния диалога, которые не связаны с предыдущими репликами и могут быть назначены начальным состоянием диалога.

Алгоритм сохраняет в состоянии все реплики собеседника, даже если это конфиденциальная информация, ее необходимо убирать в ручном режиме. Также алгоритм запоминает все ошибки, опечатки и неточности, которые были допущены оператором. Все они попадут в схему диалога и их необходимо будет исправлять. К примеру, если в диалоге фигурировало имя собеседника, то обращение в других диалогах будет также по имени, поэтому после обучения необходимо выделить изменяемые сущности и значения, получаемые извне в ручном режиме.

Положительной стороной данного алгоритма является то, что он не является черным ящиком – на него можно повлиять, представив автомат в виде схемы, состоящей из блоков, соединенных связями. Представленная особенность является решающей в выборе данного алгоритма к использованию. Также вероятность того, что алгоритм навредит обучаемой системе минимальна, так как в нем только повторяются действия оператора и весь результат обучения виден и может быть либо исправлен, либо отменен.

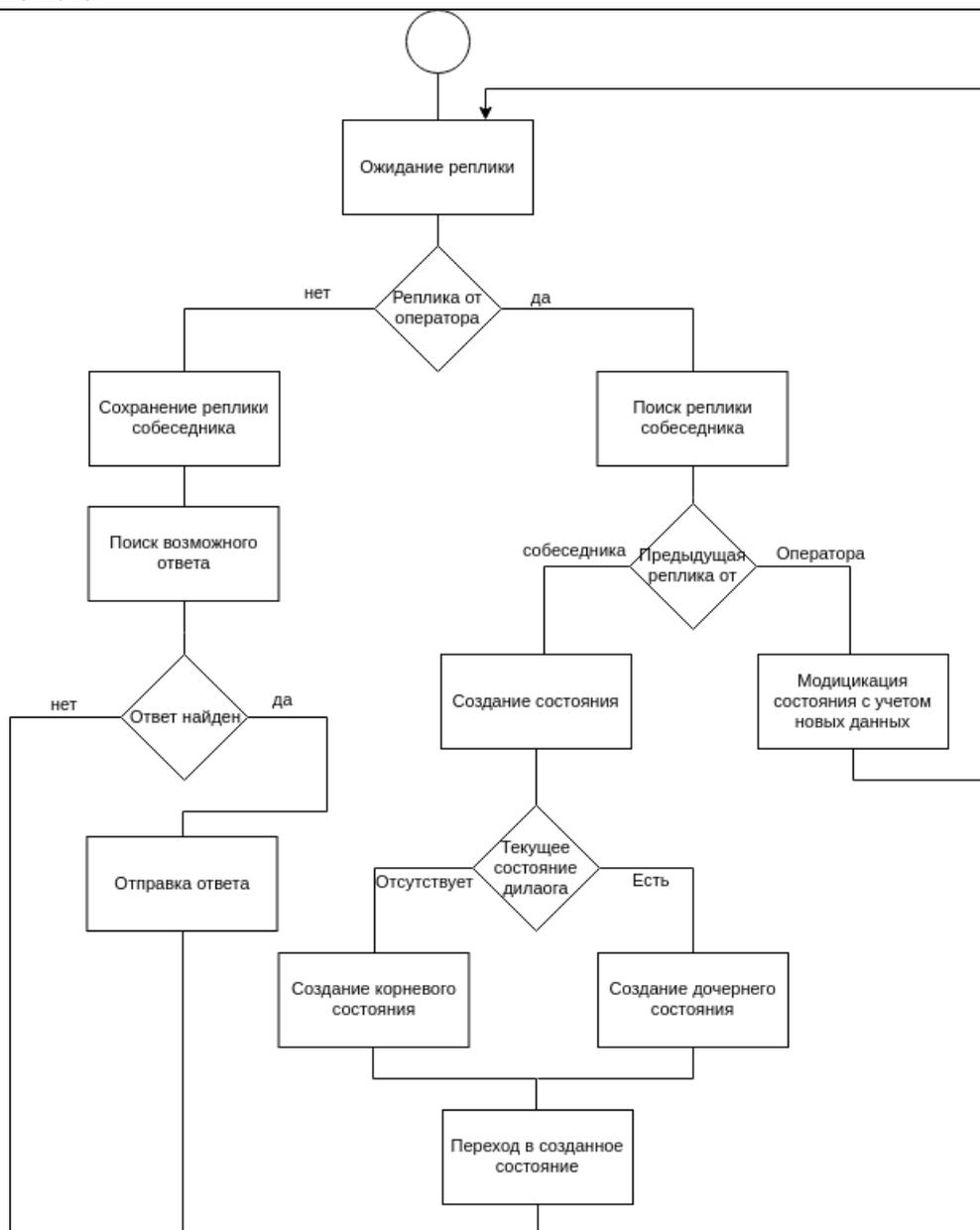


Рис. 2. Схема алгоритма обучения в режиме реального времени

Библиографический список

1. Проватар, А. И. Особенности и проблемы виртуального общения с помощью чат-ботов / А. И. Проватар, К. А. Ключко // Научные работы ВНТУ. – 2013. – № 3. – URL: <http://lib.knigi-x.ru>
 2. Автоматизированное обучение. – URL: <https://salebot.pro/articles/9>
 3. Cooper, R. J. A Simple Question Answering System / R. J. Cooper, S. M. Riiger // Imperial College of Science, Technology and Medicine. – London, 2000. – URL: <http://imperial.ac.uk>
- А. Ю. Тычков, А. К. Алимуратов, П. П. Чураков**

Пензенский государственный университет, Пенза, Россия

ОБНАРУЖЕНИЕ МАРКЕРОВ РЕЧЕВЫХ СИГНАЛОВ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ ГИЛЬБЕРТА–ХУАНГА

Обсуждаются вопросы повышения эффективности диагностики психогенных состояний по анализу речевых

сигналов (РС). Для повышения эффективности и оперативности обработки РС предложен способ определения психогенных маркеров в РС в условиях повышенного уровня помех и свободной двигательной активности человека. В основе способа предложено использовать теорию Гильберта–Хуанга (*ННТ*). Проведен анализ известной теории, выявлены ее достоинства и недостатки.

Введение

Речевой сигнал (РС) является акустическим сигналом, образуемым артикуляционным отделом речевого аппарата человека [1]. Смыслоразличительной единицей РС выступает фонема, которая служит для различения и отождествления звуков и слов [2]. РС характеризуется основным набором параметров: амплитудой, частотой и временем, представленных в сигнале в виде звукового давления, частоты основного тона и формант [3].

Известно [4–6], что пациенты, страдающие психогенными расстройствами имеют высокую восприимчивую чувствительность к внешним акустическим воздействиям в зависимости от уровня шума или звука, его громкости и скорости воспроизведения. Согласно литературным данным [7, 8], большой интерес в науке представляет изучение психогенных маркеров РС.

Основными изменениями РС, регистрируемыми в период психогенных состояний, являются: скорость речи, спектральные характеристики, интенсивность основного тона и динамика ее изменения. При психогенных расстройствах у пациентов наблюдается замедление речи, нарушение вокализации и модуляция голоса [9, 10]. Представленные изменения на медицинских сигналах врач не всегда способен правильно обнаружить. Прежде всего это связано с индивидуальными особенностями функционирования биологической сигнальной системы. Маркеры на исследуемых сигналах могут существенно различаться в зависимости от клинической картины заболевания.

Теория Гильберта-Хуанга

Для решения задач определения сигнальных маркеров психических расстройств предложено использовать теорию Гильберта–Хуанга (*Hilbert-Huang theory* или *ННТ*). В предыдущей работе авторов [11] проведен анализ методов, основывающих теорию *ННТ*: декомпозиция на амплитудно-временные составляющие (*Amplitude-time component* или *ATC*), спектральный анализ Гильберта (*Hilbert spectral analysis* или *HSA*) и их разновидности. Отметим, что первые работы с применением теории *ННТ* для обработки РС появились в 2008 году (К. Kais, М. Mola).

Метод декомпозиции на амплитудно-временные составляющие $ATC(t)$ позволяет представить исследуемый сигнал $x(t)$ в виде суммы:

$$x(t) = \sum_{i=1}^I ATC_i(t) + r(t), \quad (1)$$

где $ATC_i(t)$ – амплитудно-временные составляющие сигнала (*amplitude-time component* или *ATC*), $r(t)$ – конечная огибающая сигнала, t – отсчет времени ($0 < t \leq N$, N – число дискретных отсчетов времени в сигнале), $i = 1, 2, \dots, I$ – номер амплитудно-временной составляющей сигнала.

При проведении спектрального анализа Гильберта *HSA* для каждой *ATC* сигнала рассчитывается мгновенная частота ω , энергия которой отображается на поверхности энергетической плотности (*Surface of the energy density* или *SED*) в системе координат энергия-частота-время:

$$SED(E, f, t) = \sum_{i=1}^I ATC_i^2(t) \cdot e^{q \int \omega_i(t) dt}, \quad (2)$$

где $q = \sqrt{-1}$ – мнимая единица; ω – значение мгновенной частоты.

Теория *ННТ* имеет единственный недостаток, связанный с проблемой частотного смешивания амплитудно-временных составляющих $ATC(t)$ сигнала $x(t)$ (рис. 1). Уровень частотного смешивания рассчитывается по формуле:

$$S(F) = \frac{\sum_{t=1}^I \Delta S_i}{\sum_{t=1}^I S_i} \cdot 100\%, \quad (3)$$

где F – частотный спектр; S_i – значение площади спектра одной полосы пропускания частот; ΔS_i – значение площади перекрытия спектров полос пропускания частот, вычисляемая как:

$$\Delta S_i = \left[\sum_{j=1}^{g-1} (F_{j+1}(ATC_i(t)) - F_j(ATC_i(t))) \cdot (A(F_{j+1}(ATC_i(t))) - A(F_j(ATC_i(t)))) + \sum_{j=g}^{n-1} (F_{j+1}(ATC_{i+1}(t)) - F_j(ATC_{i+1}(t))) \cdot (A(F_{j+1}(ATC_{i+1}(t))) - A(F_j(ATC_{i+1}(t)))) \right] / 2, \quad (4)$$

при условии, что $F_i(ATC_i) = F_i(ATC_{i+1}) = F_g$.

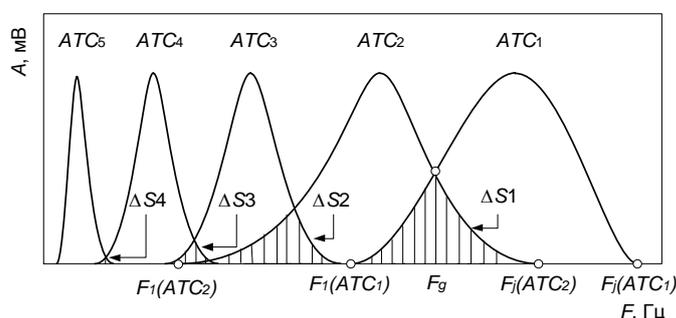


Рис. 1. Спектр полос пропускания частот амплитудно-временных составляющих исследуемых сигналов

Проведенные авторами статьи исследования теории *ННТ* на предмет определения уровня частотного смешивания $S(F)$ амплитудно-временных составляющих $ATC(t)$ сигнала $x(t)$, показали, что в $S(F)$ для РС составляет не менее 32 % при соотношении сигнал/шум не более 0,1. Чем выше уровень помех в сигнале, тем больше уровень частотного смешивания.

Данный уровень смешивания $S(F)$ амплитудно-временных составляющих $ATC(t)$ сигнала $x(t)$ является допустимым для решения задач определения психогенных маркеров в речевых сигналах. Однако, для повышения уровня помехозащищенности РС для систем мобильной диагностики, в условиях свободной двигательной активности пациента необходимо совершенствование теории *ННТ*, с целью снижения уровня смешивания ATC сигнала.

Разработка способа определения психогенных маркеров в речевых сигналах

Для определения сигнальных маркеров психогенных состояний предложен способ, блок-схема которого приведена на рисунке 2. Суть способа заключается в разложении РС на амплитудно-временные составляющие $ATC(t)$ с использованием теории *ННТ*.

Основная задача предложенного способа сводится к определению амплитудно-временных составляющих РС $ATC(t)$ с основным тоном. Процесс определения заключается в последовательном вычислении разницы Δ значений логарифмов энергии между текущей и последующей составляющими:

$$\Delta = |LE_i(ATC(t)) - LE_{i+1}(ATC(t))|. \quad (5)$$

Дальнейшее вычисление частоты основного тона РС осуществляется с использованием функции мгновенной энергии сигнала – оператора Тигра:

$$T(t) = ATC_i(t)^2 - ATC_i(t-1) \cdot ATC_i(t+1), \quad (6)$$

где $ATC_i(t) - ATC(t)$, содержащие основной тон.

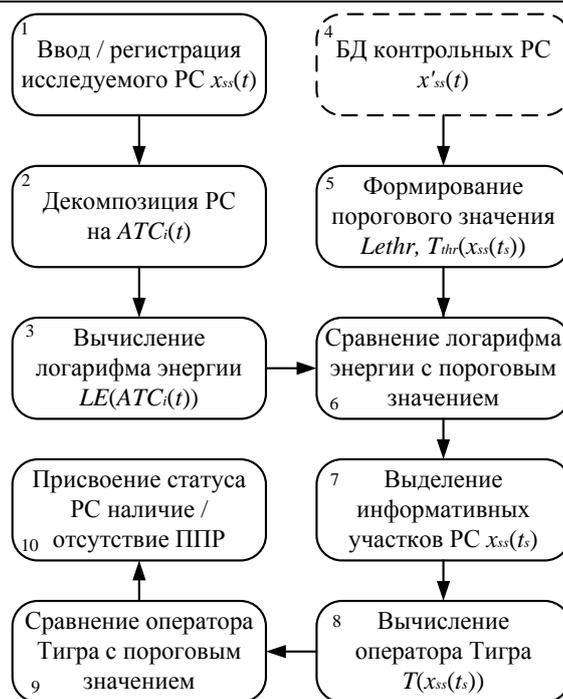


Рис. 2. Блок-схема способа определения психогенных маркеров в РС

По результатам декомпозиции РС на $ATC(t)$ вычисляется логарифм среднего значения энергии составляющих сигнала:

$$LE(ATC(t)) = \lg \sum_{t=1}^N [E(t)]^2. \quad (7)$$

Восстановление РС со средним значением логарифма энергии $LE(ATC(t))$ обеспечивает полное удаление остаточного белого шума за счет взаимного подавления пары прямых и инверсных значений.

По результатам вычисления частоты основного тона осуществляется его сравнение с пороговым значением, превышение которого говорит о наличии у пациента маркеров психогенных расстройств.

Исследование способа определения психогенных маркеров в речевых сигналах

Исследования предложенного способов выявили, что значение частоты основного тона у пациентов без патологий психического здоровья находится от 170 до 200 Гц. Однако, у большинства пациентов с психическими расстройствами значение ЧОТ находится в диапазоне от 110 до 140 Гц.

Кроме изменений частоты основного тона у пациентов с психогенными расстройствами регистрировались и другие изменения функционирования органов речевого аппарата: изменение продолжительности произношения гласных звуков (а, о); изменение значения амплитуды РС.

Обсуждения и выводы

Для оценки достоверности определения психогенных маркеров в РС использовались коэффициенты обнаруженных (*Detection rate, DR*) и необнаруженных (*Missed Detection, MD*):

$$\begin{aligned}
 DR_{ППР} &= \frac{A}{A+B} \cdot 100\%, \\
 MD_{ППР} &= \frac{B}{A+B} \cdot 100\%, \\
 DR_{ППР_{лож}} &= \frac{D}{C+D} \cdot 100\%, \\
 MD_{ППР_{лож}} &= \frac{C}{C+D} \cdot 100\%,
 \end{aligned}
 \tag{8}$$

где A – число обнаруженных сигнальных маркеров у пациентов экспериментальной группы, B – число необнаруженных сигнальных маркеров у пациентов экспериментальной группы, C – число необнаруженных сигнальных маркеров у пациентов контрольной группы, D – число обнаруженных сигнальных маркеров у пациентов контрольной группы.

Результаты расчетов показали, что сигнальный маркер – значение частоты основного тона обеспечивает значение коэффициентов DR 5 % и MD 6 %. Высокое значение коэффициента DR и MD сигнальных маркеров объясняется различной нозологией и индивидуальными особенностями исследуемых биологических сигнальных систем пациентов, даже в условиях выраженной клинической картиной заболевания.

Экспериментальные исследования подтвердили эффективность применения теории *ННТ* для решения задач технологического процесса. Для повышения достоверности принятия решения о наличии у пациента психогенных расстройств необходимо учитывать совокупность различных маркеров, регистрируемых на различных медицинских сигналах.

Разработка технологического процесса обработки РС осуществлялась при финансовой поддержке РФФИ (проект № 18-37-00256 мол_а, 2018–2019 гг.).

Библиографический список

1. Сапожков, М. А. Речевой сигнал в кибернетике и связи / М. А. Сапожков. – М. : Радио и связь, 1983. – С. 248.
2. Леонтьев, А. А. Психофизиологические механизмы речи / А. А. Леонтьев. – М. : Наука, 1970. – С. 370.
3. Михайлов, В. Г. Измерение параметров речи / В. Г. Михайлов, Л. В. Златоусова. – М. : Радио и связь, 1987. – С. 168.
4. Darby, J. K. Vocaland speech patterns of depressive patients / J. K. Darby // *Folia Phoniatrica*. – 1977. – Vol. 29. – P. 279–291.
5. Тычков, А. Ю. Идентификация личности на основе анализа речевых сигналов / А. Ю. Тычков, А. К. Алимуратов // Актуальные проблемы науки и образования : материалы VI науч. конф. студентов, аспирантов и молодых ученых. – Пенза : Изд-во ПГУ, 2012. – С. 13–17.
6. Mundt, J. Voice acoustic measures of depression severity and treatment response collected via interactive voice response (IVR) technology / J. Mundt, P. Snyder // *Journal of Neurolinguistics*. – 2007. – № 20. – P. 50–64.
7. France, D. Acoustical properties of speech as indicators of depression and suicidal risk / D. France, R. Shiavi // *IEEE Transactions on Biomedical Engineering*. – 2008. – № 47. – P. 82.
8. Tychkov, A. Yu. Program implementation of an algorithm for recognition of speech signals in the lab-view graphics programming environmen / A. Yu. Tychkov, A. K. Alimuradov, M. V. Frantsuzov, P. P. Churakov // *Measurement Techniques*. – 2015. – Vol. 58, № 9. – P. 965–969.
9. Чураков, П. П. Изучение методов анализа и обработки сигналов : учеб. пособие : в 2 ч. / П. П. Чураков, А. Ю. Тычков, А. К. Алимуратов. – Пенза : Изд-во ПГУ, 2014. – Ч. 1: Современные методы обработки речевых сигналов. – С. 72.
10. Tychkov, A. Yu. Adaptive signal processing method for speech organ diagnostics / A. Yu. Tychkov, A. K. Alimuradov, P. P. Churakov // *Measurement Techniques*. – 2016. – Vol. 59, № 5. – P. 485–490.
11. Noise-Robust Algorithm for "Speech/Pause" Segmentation in Diagnostic Systems of Psychogenic States / A. K. Alimuradov, A. Yu. Tychkov, P. P. Churakov, Yu. S. Kvitka, A. P. Zaretskiy, G. V. Vishnevskaya // 2016 International Conference on Engineering and Telecommunication (EnT) (Nov. 29–30, 2016, Moscow, Russia). – М., 2016. – P. 3–6.

А. П. Писарев, М. А. Писарев, Е. М. Голобокова

Пензенский государственный университет, Пенза, Россия

АКТУАЛЬНОСТЬ РАЗРАБОТКИ ПРОГРАММНЫХ И АЛГОРИТМИЧЕСКИХ СРЕДСТВ ДЛЯ СОЗДАНИЯ ПРИБОРОВ ПРОГНОЗИРОВАНИЯ ИНФАРКТА МИОКАРДА

Рассмотрена актуальность разработки программных и алгоритмических средств для создания и совершенствования приборов прогнозирования инфаркта миокарда, с целью определения наступления критических состояний, что важно для оценки сердечно-сосудистой и других систем организма.

Заболевания сердечно-сосудистой системы (ССС) являются в настоящее время главной причиной смертности населения экономически развитых стран. С каждым годом частота и тяжесть этих заболеваний возрастают, что является главной причиной смертности среди лиц трудоспособного возраста, в этой связи возрастает потребность в поиске новых решений для ранней диагностики ишемических изменений миокарда.

Форма пульсовой волны индивидуальна, до сих пор не существует ее полной клинической интерпретации. Соответственно любые изменения в функционировании сердца также индивидуальны. В связи с этим не существует эталона для диагностирования инфаркта по форме пульсовой волны [1]. Инфаркт миокарда – одна из клинических форм ишемической болезни сердца, ее причинами являются заболевания вызывающие отклонения в функционирования сердечно-сосудистой системы. Мониторируя эти отклонения можно с высоким уровнем точности определить наступления критических состояний, в том числе и инфаркта миокарда. Определить степень изменений в работе сердечно-сосудистой системы можно проведя контурный анализ и сравнение с моделью, имитирующей поведение артерии во время ее функционирования. В связи с этим ставится задача создания имитационной модели кровотока, а также алгоритма определения отклонений и их корреляции с изменением параметров математической моделью.

Подобный подход дает возможность делать прогнозы по профилактике различных видов заболеваний и критических состояний в том числе и инфаркта миокарда. Анализ механизмов, лежащих в основе этого заболевания, оказывается неизбежно связан с формулировании теоретических и биологических моделей наблюдаемой динамики. Подобные модели обладают такими свойствами как автоколебания или «хаос». Их моделирование в отсутствие конкретных применений к клиническим или экспериментальным системам остается на уровне гипотез. В виду чего возникает необходимость непрерывного взаимодействия между теорией и экспериментом [2].

В настоящий момент благодаря развитию технологии передачи, обработки, отображения информации на базе существующих устройств, таких как: пульсометр, беспроводной интернет, датчиков измерения артериального давления, скорости кровотока, пульса, появилась возможность существенно расширить функции существующих систем мониторинга состояния сердечно-сосудистой системы (ССС). Эффект достигается за счет увеличения времени наблюдения и объема измерительной информации о параметрах пульсирующего функционирования кровеносной системы. Это открывает возможность повысить качество диагностики ранних стадий заболеваний, которые могут быть выявлены только при анализе данных полученных за продолжительное время [3]. Для их выявления используют сложные математические методы статистического анализа и моделирования, которые в настоящее время развиваются. Увеличивается количество публикаций и экспериментальных исследований в данной области. Существенно развивается и техническая база реализации этого направления за счет развития микроэлектроники. Соответственно актуальным является разработка программных и алгоритмических средств для создания приборов медицинского назначения в этой области.

Сложность математических методов является следствием сложности объекта исследования так как большинство процессов в организме человека являются нелинейными. Как правило используются имитационные математические модели и численные методы расчетов являющимися наиболее эффективными способами решения нелинейных задач.

Кровеносная система человека является активной средой в которой важное место занимает нервная регуляция, которая за счет биологической обратной связи обеспечивает гомеостаз внутренних органов. Поэтому для корректного описания наблюдаемых процессов кровотока необходимо учитывать активность среды. Доказано, что энергия которая необходимая для перекачивания крови распределяется между работой сердечной мышцы и сосудов коронарной системы в соотношении один к трем. Прохождение крови в капилляров в зависимости от регулирующих воздействий меняется на порядок, а в капиллярах верхнего слоя на два десятичных порядка [4]. Очевидно, что эти эффекты биологической обратной связи необходимо учитывать при описании модели кровотока. Только в этом случае можно получить адекватную математическую модель.

Достоинством такого подхода является то, что любое отклонение в параметрах сигналов, снимаемых с датчиков измеряющих биомедицинские сигналы, может быть описано путем изменения параметров математической модели. В свою очередь изменение параметров математической модели позволяет предположить с высокой степенью достоверности о изменении физиологических параметров биологической модели (человека). При длительном наблюдении за такими отклонениями можно построить статистическую имитационную модель физиологического состояния пациента, которая позволяет выявить наличие артефактов, т.е. отклонений от среднестатистических показателей, которые можно интерпретировать как предвестники начала заболеваний. А на следующем этапе в сочетании с амбулаторными исследованиями дает более полную картину необходимую для диагностики ранних стадий заболеваний.

Для создания имитационной модели необходимо описать процессы пульсирующего функционирования кровеносной системы. Физически процесс работы кровеносной системы представлен на рис. 1.

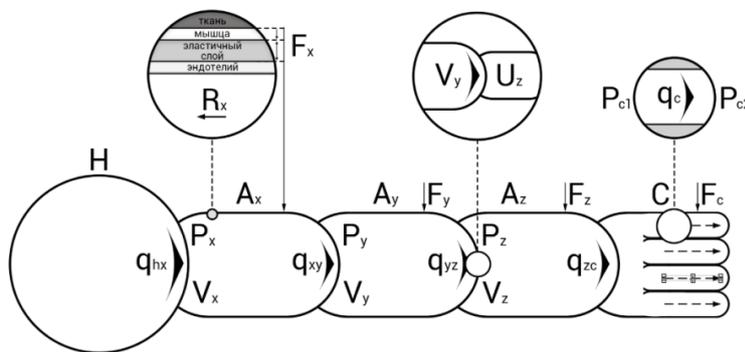
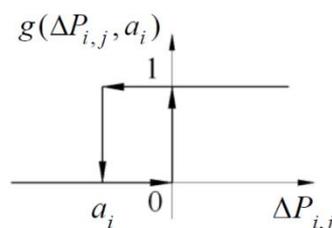


Рис. 1. Механика работы кровеносной системы

Сосуд представлен в виде однородных камер (в физическом представлении являющихся мягкой, заполненной вязкой жидкостью, трубкой) соединяющих источник давления жидкости (С) и систему капилляров (В), которые имитируют наличие упругой оболочки, также передвижение объема крови благодаря наличию разности давления на входе и выходе каждого элемента, а также наличия силы вязкости препятствующего кровотоку [5].

Процесс кровотока протекает следующим образом: синхронно с открытием клапана источника давления (сердца) под воздействием нервной системы происходит расслабление мышц и снижение внешнего давления на камеру. Поступающая жидкость, ввиду сильного трения и отсутствия внешнего давления начинает расширять камеру. В момент закрытия клапана происходит мышечное сжатие, что приводит к увеличению давления на стенки камеры, и поступившая жидкость выталкивается как в следующую камеру, так и сторону клапана, образуя обратную волну. Математическая модель работы сердца может быть описана в первом приближении уравнением гистерезиса в соответствии с рис. 2 [5].

Рис. 2. Гистерезисная функция $g(\Delta P_i, j, a_i)$

Клапан сердца открывается, в момент возникновения перепада давления в направлении открытия, как показано в условии (1)

$$P_i - 1(t) - P_j(t) > 0. \quad (1)$$

Клапан остается открытым, пока выполняется условие (2)

$$q_i(t) > -q_i^*. \quad (2)$$

Если обратный поток превысит заданное значение q_i , клапан закроется. Таким образом задача создания имитационной модели кровотока, а также алгоритма определения отклонений и их корреляции с изменением параметров математической моделью представляется важной и актуальной, что предоставит возможность делать прогнозы по профилактике различных видов заболеваний и критических состояний, в том числе и инфаркта миокарда.

Библиографический список

1. Усанов, Д. А. Методы и аппаратура для диагностики состояния сердечно-сосудистой системы по характеристикам пульсовой волны / Д. А. Усанов, А. В. Скрипаль, А. Ю. Вагарин, А. П. Рытик. – Саратов : Изд-во Саратов. ун-та, 2009. – 96 с.
2. Писарев, А. П. Математическое описание поведения артерии при внешнем компрессионном воздействии / А. П. Писарев, М. А. Писарев // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. III Ежегодной межвуз. студ. науч.- практ. конф. – Пенза, 2016. – С. 128–132.
3. Писарев, А. П. Прогнозирование заболеваний с использованием хронобиологических данных / А. П. Писарев, М. А. Писарев, Е. М. Голобокова // Актуальные проблемы гуманитарных и естественных наук. – 2017. – № 11–3. – С. 35–37.
4. Быков, И. В. Принципы построения математической модели для исследования взаимодействия насосов непрерывного потока и сердечно-сосудистой системы / И. В. Быков, Г. П. Иткин // Вестник трансплантологии и искусственных органов. – 2013. – Т. 15, № 3. – С. 59–65.
5. Фролов, С. В. Модель сердечно-сосудистой системы, ориентированная на современную интенсивную терапию / С. В. Фролов, С. Н. Маковеев, Д. Ш. Газизова, В. А. Лищук // Вестник ТГТУ. – 2008. – Т. 14, № 4. – С. 892–901.

А. П. Писарев, М. А. Писарев, Е. М. Голобокова

Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ АДАПТИВНЫХ И НЕЛИНЕЙНЫХ ФИЛЬТРОВ ПРИ ПОЛУЧЕНИИ И ОБРАБОТКЕ КОМПОНЕНТ ПУЛЬСОВОЙ ВОЛНЫ

Рассмотрены проблемы разработки эффективных методик цифровой обработки сигнала артериальной пульсации крови в условиях воздействия широкополосных помех различной интенсивности с целью повышения эффективности детектирования пульсовой волны и определения ее амплитудно-временных характеристик с наименьшей погрешностью, что важно для оценки сердечно-сосудистой и других систем организма.

Артериальное давление (АД) является одним из основных показателей состояния здоровья человека, поэтому его измерение является частью полноценного диагностического обследования [1].

Известно, что в течение сердечного цикла давление колеблется между максимальным (систолическим) и минимальным (диастолическим) значением. Величина давления связана с работой сердца и сопротивлением сосудов и уменьшается по мере того, как кровь отходит от сердца через артерии, и она снижается в мелких артериях и артериолах и продолжает падать по мере прохождения крови через вены и капилляры обратно к сердцу [2]. Таким образом величина АД и форма пульсовой волны являются результатом изменений вызванных эндогенными процессами организма, являясь основным интегральным показателем его деятельности. АД влияет на гемолимфонаполнение сосудов и сопровождает жизненно важные процессы во всех функциональных системах: кровообращения, терморегуляции, дыхания и других.

Отклонения величины АД и формы пульсовой волны от условно нормальных значений говорят о патологических изменениях, что важно для оценки сердечнососудистой и других систем организма.

Регистрация артериальной пульсации крови используется в кардиологической диагностике, а также в системах неинвазивного мониторинга артериального давления. Для принятия диагностических решений при автоматической обработке формы пульсовой волны необходима информация о длительностях периодов компонент, форме импульсов, тонкой структуре их сигнала [3, 4].

Сигнал кровяного давления состоит из нескольких компонент, прямая волна, обратная волна, волна дыхания (см. рисунок 1).

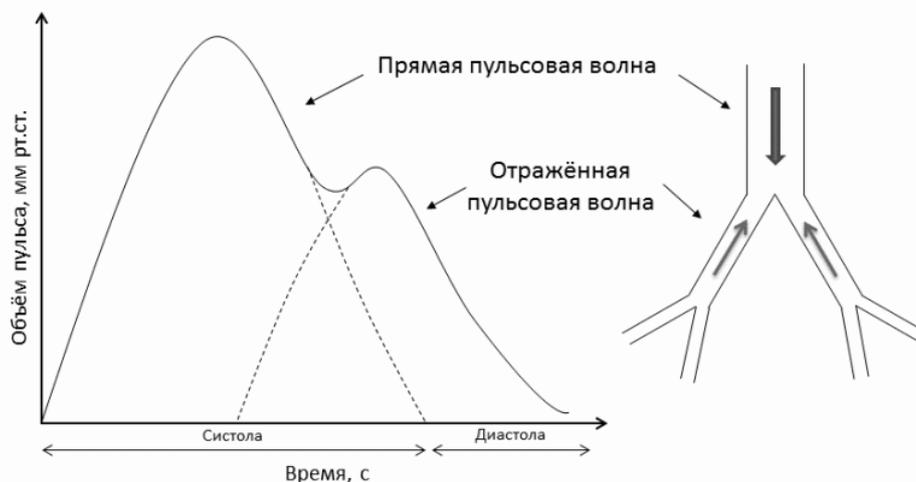


Рис. 1. Артериальная пульсация крови

Компоненты пульсовой волны квазипериодичны. Однако между ними в разных периодах существует высокий уровень корреляции. Важным является присутствие в пульсовой волне диагностически значимых хаотических колебаний, являющиеся информативной тонкой структурой сигнала. Эффективность их обнаружения и точность измерения амплитудно-временных параметров пульсовой волны во многом определяет погрешность определения диагностических показателей, таких как частота сердечных сокращений и сатурация гемоглобина крови кислородом.

Таким образом, актуальной задачей является разработка эффективных методик цифровой обработки сигнала артериальной пульсации крови в условиях воздействия широкополосных помех различной интенсивности с целью повышения эффективности детектирования пульсовой волны и определения ее амплитудно-временных характеристик с наименьшей погрешностью.

Поскольку помехи имеют физическую природу отличную от процессов формирования пульсовой волны то, как показывает практика помехи практически не коррелируют как

между собой, так и с информативным сигналом. Возникновение помех в измерительных приборах может быть связано с флуктуациями физических величин вызванных сетевыми помехами, атмосферными явлениями, электромагнитными и электростатическими полями, и другими явлениями, возникающими в процессе работы прибора [5,6].

Существуют также и помехи связанные с мышечной активностью человека. В процессе движения происходит сокращение мышц, вызванное электрической активностью внутри мышцы, учащается сердцебиение и дыхание. Все эти факторы отрицательно сказываются на вариабельности сигнала вызывая значительное изменение формы волны. На рис. 2 показана пульсовая волна на фоне помех, наиболее распространенных видов: наводка напряжения сети, дрейф изолинии при изменении напряжения поляризации, помехи вызванные движениями пациента, потенциалы мышц.

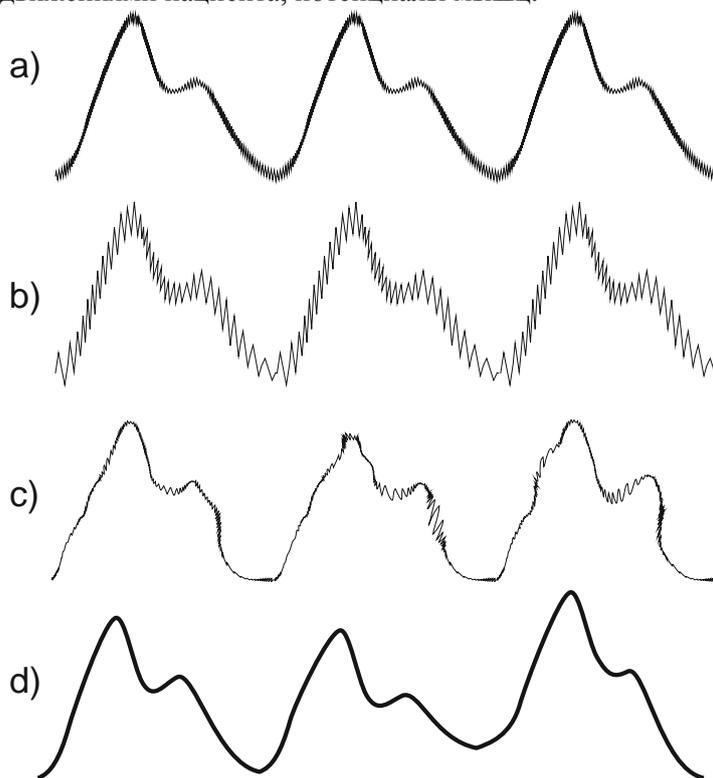


Рис. 2. Распространенные типы помех пульсовой волны.
a – потенциалы мышц; *b* – сетевая помеха; *c* – шумы движения; *d* – дрейф изолинии

Методы повышения помехозащищенности при получении и обработке пульсовой волны можно разделить на две группы. К первой относятся методы снижения помех в процессе получения сигнала:

- снижение и стабилизация межэлектродного сопротивления;
- совершенствование конструкции электродов и средств их крепления;
- выбор специальных отведений с малым уровнем помех;
- разработка помехозащищенных способов связи электродов с усилителем.
- электрическое и магнитное экранирование;
- использование сетевых фильтров, уменьшающих распространение по проводам кондуктивных помех от их источников;
- симметрирование измерительной цепи так, чтобы на оба дифференциальных входа действовали бы одинаковые помехи.

Не смотря на это уровень помех, при длительном исследовании пульсовой волны в режиме двигательной активности, может быть достаточно велик. В виду этого применяется и вторая группа методов повышения помехозащищенности. Эти методы базируются на использовании различий между полезным сигналом и помехой. В отличие от методов защиты они называются методами повышения помехоустойчивости. К ним относятся:

- предварительная частотная фильтрация ЭКС режекторные, адаптивные или неадаптивные фильтры, ФНЧ, ФВЧ;
- селекция неинформативных элементов сигнала;
- автоматическая регулировки уровня сигнала;
- автоматическая центровка формы импульса;
- пространственно-временная обработка;
- использование дифференциальных усилителей.

В случаях, когда подавление помех оказывается неэффективным (т.е. уровень шумов превышает уровень сигнала), зашумленные участки сигнала исключают из анализа [7, 8].

Алгоритм удаления помех должен изменять свои параметры в зависимости от меняющихся условий среды, в противном случае он может использоваться лишь для определенного формата данных (постоянная частота дискретизации), а также для заданного уровня и состава помех.

В связи с этим предложен метод помехоустойчивой обработки основанный на использовании адаптивных и нелинейных фильтров (см. рис. 3).

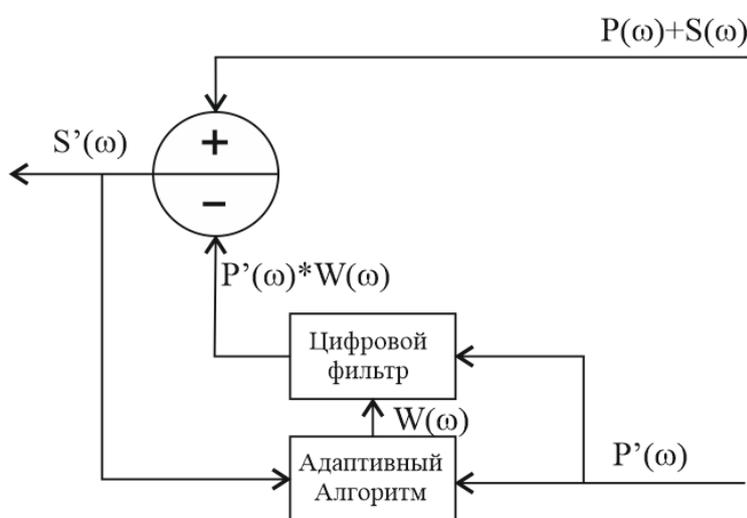


Рис. 3. Адаптивный двухканальный фильтр

На фильтр одновременно подается зашумленный сигнал $P_{(\omega)} + S_{(\omega)}$ и коррелированный с $P_{(\omega)}^*$ сигнал $P_{(\omega)}$. $W_{(\omega)}$ коэффициент используется для формирования оценки сигнала $P_{(\omega)}$. Полезный сигнал оценивается по разности (1):

$$\|P_{(\omega)}^* * W_{(\omega)} - P_{(\omega)}\| \rightarrow \min \quad (1)$$

$$P_{(\omega)}^* = P_{(\omega)}$$

$$\|S_{(\omega)} + P_{(\omega)} - W_{(\omega)} * P_{(\omega)}^*\| \rightarrow \min \quad (2)$$

Адаптивные фильтры позволяет исключить помеху из измерительного канала путем адаптивной подстройки с минимальными потерями информации в измерительной части сигнала. Точность оценки шума можно увеличить, используя в качестве его источника второй датчик, не подверженный колебаниям давления в артерии, например на другой стороне руки [9].

Даже при высокой эффективности адаптивных фильтров помехи возникающие при двигательной активности будут оказывать значительное влияние на форму сигнала. Для устранения подобных помех в измерительных приборах можно использовать нелинейные фильтры. В таком случае представляется возможным выбрать из множества измерительных сигналов те, которые не подверглись дестабилизирующим факторам. Выборку можно осу-

шестить, используя дополнительный датчик к примеру, акселерометр. Периоды, полученные в момент превышения определенного уровня сигнала с акселерометра необходимо отбросить как некорректные. Основываясь на предложенных подходах, предлагается структурная схема помехозащищенного плетизмографа (см. рис. 4).

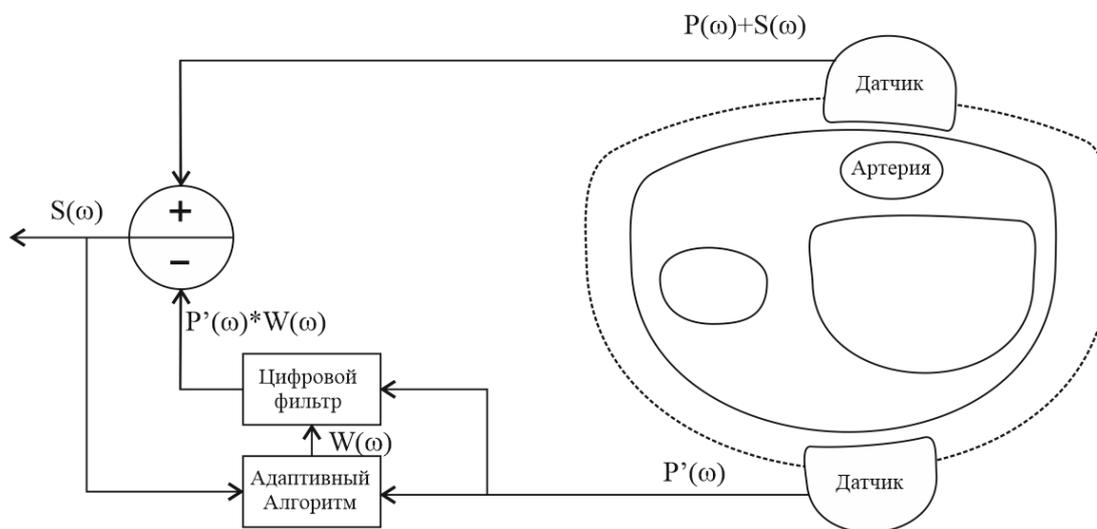


Рис. 4. Нелинейный фильтр с акселерометром

Предложенный метод позволяет снизить уровень помех до допустимого, отбросить некорректные сигналы и при этом сохранить тонкую структуру сигнала пульсовой волны, что повысит эффективность детектирования пульсовой волны для определения ее амплитудно-временных характеристик с наименьшей погрешностью.

Библиографический список

1. Human Blood Pressure Determination by Sphygmomanometry / Dorothee Perloff, Carlene Grim, John Flack, Edward D. Frohlich, Martha Hill, Mary McDonald, Bruce Z. Morgenstern // AHA Medical/Scientific Statement. Special Report. – URL: <http://aafp.org>
2. Klabunde, R. Cardiovascular Physiology Concepts / R. Klabunde. – Philadelphia : Lippincott Williams & Wilkins, 2015. – P. 93–94.
3. Агаджанян, Н. А. Биоритмы, спорт, здоровье / Н. А. Агаджанян, Н. Н. Шабатура. – М. : Физкультура и спорт, 1989. – 207 с.
4. Арушанян, Э. Б. Хронотропная активность лекарств / Э. Б. Арушанян / Экспериментальная и клиническая фармакология. – 2002. – № 1. – С. 75–76.
5. Писарев, А. П. Прогнозирование заболеваний с использованием хронобиологических данных / А. П. Писарев, М. А. Писарев, Е. М. Голобокова // Актуальные проблемы гуманитарных и естественных наук. – 2017. – № 11-3. – С. 35–37.
6. Волобуев, А. Н. Циркадианная биофизика и хрономедицина / А. Н. Волобуев, В. Ф. Пятин, Н. П. Романчук // Здоровье и образование в XXI веке. – 2016. – Т. 18, № 5. – С. 97–99.
7. Рапопорт, С. И. Хрономедицина, циркадианные ритмы / С. И. Рапопорт / Клиническая медицина. – 2012. – № 8. – С. 73–75.
8. Хильдебрандт, Г. Хронобиология и хрономедицина / Г. Хильдебрандт, М. Мозер, М. Лехофер. – М. : Арнебия, 2006. – 145 с.
9. Писарев, А. П. Математическое описание поведения артерии при внешнем компрессионном воздействии / А. П. Писарев, М. А. Писарев // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. III Ежегодной межвуз. студ. науч.-практ. конф. – Пенза, 2016. – С. 128–132.

Ярославский государственный университет им. П. Г. Демидова, Ярославль, Россия

УСОВЕРШЕНСТВОВАННЫЙ НЕЙРОСЕТЕВОЙ АЛГОРИТМ ПОДАВЛЕНИЯ ШУМА В РЕЧЕВЫХ СИГНАЛАХ НА ОСНОВЕ МЯГКОЙ МАСКИ

Продолжение исследований по разработке алгоритма шумоподавления на основе мягкой маски. Для ее оценки усовершенствованный алгоритм использует рекуррентную нейронную сеть и учитывает априорную информацию о спектральной плотности мощности шума в каждой частотной полосе.

Введение. Проблема восстановления речевого сигнала, искаженного аддитивным некоррелированным шумом, в случае, когда доступен только зашумленный сигнал, широко изучалась в прошлом и актуальна сейчас [1, 2]. Предложены методы подавления шума в частотной области, использующие различные функции коррекции спектра (ФКС), зависящие от апостериорного отношения сигнал/шум (ОСШ) и/или оценки априорного ОСШ [2–5]. В качестве ФКС также могут использоваться бинарные [6, 7] и мягкие маски [8–10].

С развитием эффективных методов машинного обучения широкое распространение стали получать алгоритмы подавления шума на основе искусственных нейронных сетей [11, 12]. Данные алгоритмы предварительно обучаются на большом наборе речевых сигналов и могут использовать дополнительную априорную информацию, такую как, например, видеозапись говорящего [13].

Данная работа является продолжением исследований по разработке нейросетевого алгоритма шумоподавления на основе мягкой маски [14, 15]. Для ее оценки усовершенствованный алгоритм использует рекуррентную нейронную сеть; кроме того, он учитывает априорную информацию о спектральной плотности мощности шума в каждой частотной полосе.

Целью работы является повышение качества речевых сигналов в условиях воздействия аддитивных шумов различных типов с помощью алгоритма шумоподавления на основе рекуррентной нейронной сети.

База речевых сигналов. В работе используется база речевых сигналов, составленная на основе записей речевого корпуса CHAINS школы компьютерных наук и информатики университетского колледжа Дублина [16]. Для обучения нейронной сети использованы записи 30 дикторов (по 4 записи на диктора), общей продолжительностью 1 час 17 минут 29 секунд. Тестирование производилось на записях оставшихся 6 дикторов (по 34 записи на диктора), общей продолжительностью 9 минут 12 секунд.

Нейросетевой алгоритм шумоподавления. В работе речевые сигналы предварительно разбивались на перекрывающиеся окна длиной 32 мс, что при частоте дискретизации 8 кГц соответствует 256 отсчетам. Величина перекрытия окон составляла 50 % от длины окна. При получении спектра сигнала использовалось окно Хеннинга.

Предлагаемая рекуррентная нейронная сеть имеет 2 входа. На вход № 1 подается текущее обрабатываемое окно с амплитудным спектром сигнала, а также 31 предыдущее. На вход № 2 – апостериорное отношение сигнал/шум текущего окна [8]. После входа № 1 следуют 2 последовательно идущих управляемых рекуррентных блока (Gated Recurrent Unit, GRU) [17] со 129 нейронами и функцией активации на основе гиперболического тангенса. После входа № 2 следует плотный слой со 129 нейронами и функцией активации на основе гиперболического тангенса. Выходы описанных цепочек скрытых слоев объединяются в одну путем конкатенации. Далее следует цепочка из 5 последовательно идущих плотных слоев с 258 нейронами и функцией активации ELU (Exponential Linear Unit) [18]. На выходе нейронной сети расположен плотный слой со 129 нейронами и логистической функцией активации.

В качестве целевой переменной для обучения выступает мягкая маска, которая является оценкой вероятности того, что в данной точке частотно-временного представления сигнала содержится преимущественно речь, а не шум. Для обучения нейронной сети сигналы искажались аддитивным белым гауссовским шумом (АБГШ), а также шумами «Speech babble» (далее – SB) и «Vehicle interior noise» (далее – VIN) из библиотеки NOISEX-92 [19] для достижения ОСШ от 0 до 20 дБ. Тип шума и ОСШ для каждого сигнала выбирался случайным образом. После искажения сигналы подавались на входы нейронной сети для ее обучения. После этого данная процедура повторялась, причем тип шума и ОСШ вновь выбирались случайно для каждого сигнала, что обеспечило очень большой набор вариаций сигнал – тип шума – ОСШ. Произведено 1000 эпох обучения (1 эпоха – полный набор сигналов обучающей выборки).

Исследование качества работы алгоритма. Проведено сравнение предложенного алгоритма шумоподавления с широко используемыми в задаче повышения качества зашумленных речевых сигналов. Для удобства обозначим участвующие в сравнении алгоритмы следующим образом:

1) алгоритм 1 – предложенный алгоритм, основанный на оценке мягкой маски с использованием рекуррентной нейронной сети;

2) алгоритм 2 – двухступенчатый алгоритм шумоподавления ($\alpha = 0,98$) на основе ФКС Винера;

3) алгоритм 3 – алгоритм на основе подхода прямого принятия решения ($\alpha = 0,98$) и ФКС Винера.

В ходе исследования тестовые сигналы искажались АБГШ, а также шумами SB и VIN для достижения ОСШ 0, 5, 10, 15, 20 дБ. Затем производилась обработка сигналов с помощью указанных алгоритмов шумоподавления.

Эффективность работы алгоритмов оценивалась с помощью показателя качества PESQ (Perceptual Evaluation of Speech Quality) [20]. Оценка качества производилась для каждого восстановленного сигнала, затем результаты усреднялись. Результаты проведенного сравнения представлены в табл. 1.

Таблица 1

Сравнение алгоритмов шумоподавления с использованием PESQ

| Тип шума | АБГШ | | | SB | | | VIN | | |
|----------|------|------|------|------|------|------|------|------|------|
| | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| ОСШ, дБ | | | | | | | | | |
| 0 | 2,40 | 1,97 | 2,06 | 2,11 | 1,37 | 1,66 | 3,82 | 3,56 | 3,67 |
| 5 | 2,70 | 2,28 | 2,41 | 2,58 | 1,89 | 2,11 | 4,05 | 3,84 | 3,92 |
| 10 | 3,00 | 2,55 | 2,77 | 2,97 | 2,38 | 2,55 | 4,22 | 4,11 | 4,14 |
| 15 | 3,31 | 2,86 | 3,11 | 3,31 | 2,82 | 2,96 | 4,33 | 4,27 | 4,28 |
| 20 | 3,60 | 3,21 | 3,40 | 3,61 | 3,25 | 3,37 | 4,41 | 4,37 | 4,38 |

Видно, что предложенный алгоритм обеспечивает лучшее качество обработанного сигнала по показателю качества PESQ для всех используемых типов шума по сравнению с двумя другими алгоритмами. Так для АБГШ преимущество предложенного алгоритма (алгоритм 1) перед алгоритмом на основе подхода прямого принятия решения и ФКС Винера (алгоритм 3) составляет 0,2–0,35 (в среднем 0,25) единицы шкалы MOS (Mean Opinion Score). Для шума SB – 0,25–0,46 (в среднем 0,38); для шума VIN – 0,03–0,15 (в среднем 0,09).

Отметим, что в случае с шумом VIN алгоритмы обеспечивают близкие значения показателя качества PESQ. Это можно объяснить особенностью шума, энергия которого сосредоточена в области низких частот. Таким образом, значительная часть диапазона частот, соответствующего речевым сигналам, в данном случае практически не искажается, и на итоговую оценку влияет в том числе и способность алгоритма обрабатывать практически неискаженные спектральные составляющие речи.

Заключение. В работе предложена усовершенствованная методика оценки мягкой маски с использованием рекуррентной нейронной сети и алгоритм подавления шума в ре-

чевых сигналах на ее основе. С использованием показателя качества PESQ проведено сравнение разработанного алгоритма с двумя широко известными алгоритмами шумоподавления, использующими ФКС Винера. Результаты сравнения демонстрируют преимущество предлагаемого алгоритма и свидетельствуют о его перспективности.

Библиографический список

1. Шумоподавление на основе перцептуальных алгоритмов спектрального вычитания и обработки сигналов в подпространствах / А. А. Петровский, И. С. Азаров, Д. С. Лихачёв, Ю. Н. Ромашкин, Ю. Н. Жигулёвцев, А. А. Харламов // *Речевые технологии*. – 2012. – № 4. – С. 4–15.
2. Scalart, P. Speech enhancement based on a priori signal to noise estimation / P. Scalart, J. V. Filho // 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings. – 1996. – Vol. 2. – P. 629–632.
3. Lim, J. Enhancement and bandwidth compression of noisy speech / J. Lim, A. Oppenheim // *Proceedings of the IEEE*. – 1979. – Vol. 67, № 12. – P. 1586–1604.
4. Boll, S. Suppression of acoustic noise in speech using spectral subtraction / S. Boll // *IEEE Transactions on Acoustics, Speech, and Signal Processing*. – 1979. – Vol. 27, № 2. – P. 113–120.
5. Ephraim, Y. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator / Y. Ephraim, D. Malah // *IEEE Transactions on Acoustics, Speech, and Signal Processing*. – 1984. – Vol. 32, № 6. – P. 1109–1121.
6. Wang, D. On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis / D. Wang // *Speech Separation by Humans and Machines*. – Boston : Kluwer Academic Publishers, 2005. – P. 181–197.
7. Тупицин, Г. С. Предобработка зашумленных речевых сигналов с помощью бинарных масок в задаче идентификации диктора / Г. С. Тупицин, А. И. Топников, А. Л. Приоров // *Наукоемкие технологии*. – 2015. – № 11. – С. 56–61.
8. Tupitsin, G. Two-step noise reduction based on soft mask for robust speaker identification / G. Tupitsin, A. Topnikov, A. Priorov // 2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT). – 2016. – P. 351–356. – URL: <http://ieeexplore.ieee.org>
9. Lu, Y. Estimators of the Magnitude-Squared Spectrum and Methods for Incorporating SNR Uncertainty / Y. Lu, P. C. Loizou // *IEEE Transactions on Audio, Speech, and Language Processing*. – 2011. – Vol. 19, № 5. – P. 1123–1137.
10. Тупицин, Г. С. Методика оценки мягкой маски для задачи предобработки зашумленных речевых сигналов в системах идентификации диктора / Г. С. Тупицин, А. И. Топников, А. Л. Приоров // *Успехи современной радиоэлектроники*. – 2016. – № 6. – С. 73–80.
11. Zhao, H. Convolutional-Recurrent Neural Networks for Speech Enhancement / H. Zhao, S. Zarar, I. Tashev, C. Lee // 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. – 2018. – P. 2401–2405. – URL: <http://ieeexplore.ieee.org>
12. Kolbaek, M. Monaural Speech Enhancement using Deep Neural Networks by Maximizing a Short-Time Objective Intelligibility Measure / M. Kolbaek, Z. Tan, J. Jensen. – 2018. – URL: <http://dl.acm.org>
13. Audio-Visual Speech Enhancement Based on Multimodal Deep Convolutional Neural Network / J. Hou, S. Wang, Y. Lai, Y. Tsao, H. Chang, H. Wang. – URL: <https://arxiv.org/abs/1703.10893>
14. Тупицин, Г. С. Методика оценки мягкой маски с использованием сверточных нейронных сетей для подавления шума в речевых сигналах / Г. С. Тупицин, А. И. Топников // *Радиоэлектронные средства получения, обработки и визуализации информации : сб. докл. 7-й Всерос. конф.* – М., 2017. – С. 137–141.
15. Свидетельство о государственной регистрации программы для ЭВМ № 2018614631 от 11 апреля 2018 г. ConvNN speech denoiser – программа для восстановления искаженных речевых сигналов / Г. С. Тупицин, А. И. Топников.
16. Cummins, F. The CHAINS Speech Corpus: CHAracterizing INdividual Speakers / F. Cummins, M. Grimaldi, T. Leonard, J. Simko // *Proc. of SPECOM*. – S. Petersburg, 2006. – P. 1–6.
17. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation / K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio // *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. – Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. – P. 1724–1734.
18. Clevert, D.-A. Fast and Accurate Deep Network Learning by Exponential Linear Units / D.-A. Clevert, T. Unterthiner, S. Hochreiter. – URL: <https://arxiv.org/abs/1511.07289>
19. Varga, A. Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems / A. Varga, H. J. M. Steeneken // *Speech Communication*. – 1993. – Vol. 12, № 3. – P. 247–251.

20. International Telecommunication Union. P. 862: Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs / International Telecommunication Union // ITU-T Recommendation. – 2001. – Vol. 862. – P. 862.

А. К. Алимуратов, А. Ю. Тычков, П. П. Чураков

Пензенский государственный университет, Пенза, Россия

СПОСОБ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОЦЕНКИ ПСИХОЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ОПЕРАТОРОВ СИСТЕМ УПРАВЛЕНИЯ С ВЫСОКОЙ СТЕПЕНЬЮ ОТВЕТСТВЕННОСТИ

Предложен способ обработки речевых команд подсистем голосового управления, повышающий эффективность оценки психоэмоционального состояния операторов систем управления с высокой степенью ответственности (медицинских экзоскелетов), заключающийся в использовании адаптивного метода обработки нестационарных сигналов, путем декомпозиции на эмпирические моды, и определении набора информативных параметров, релевантных психоэмоциональной составляющей речевых команд.

Корректная оценка психоэмоционального состояния человека на сегодняшний день является важной социально-значимой государственной проблемой, поскольку напрямую связана с формированием здорового образа жизни населения и особенно важна в тех отраслях человеческой деятельности, сопряженных с повышенным риском для жизни. К системам управления с высокой степенью ответственности относят биоинженерные робототехнические устройства – медицинские экзоскелеты, управляемые с помощью речевых команд [1–3].

На сегодняшний день на рынке речевых технологий представлены коммерческие системы анализа голоса для оценки психоэмоционального состояния человека. Наибольший интерес в данных системах представляют способы и применяемые в них методы обработки речевых команд. Однако в силу коммерческой тайны производители не предоставляют такую информацию.

Целью данной работы является разработка способа обработки речевых команд, повышающего эффективность оценки психоэмоционального состояния операторов медицинских экзоскелетов. Исследование является развитием ранее опубликованных трудов авторов [3, 4] и выполнено при финансовой поддержке Совета по грантам Президента РФ в рамках проекта № СП-246.2018.5.

Проведенный обзор известных способов обработки речевых сигналов [5, 6] выявил, что для повышения эффективности оценки психоэмоционального состояния, необходимо: использовать адаптивные методы обработки сложных, нестационарных речевых сигналов; использовать информативные параметры речевых команд релевантные психоэмоциональной составляющей.

На рис. 1 представлена блок-схема способа обработки речевых команд в составе модуля оценки психоэмоционального состояния интеллектуальной подсистемы голосового управления медицинскими экзоскелетами. Суть способа заключается в предварительной обработке (блоки 1-3), определении информативных параметров на основе энергетического, кепстрального, просодического и формантного анализов (блоки 5-8), предварительно разложив речевую команду на составляющие с помощью адаптивного метода обработки – декомпозиции на эмпирические моды (ДЭМ) (блок 4).

Декомпозиция на эмпирические моды. ДЭМ – это адаптивная технология разложения нестационарных сигналов нелинейных систем [7]. ДЭМ обеспечивает локальное

разложение сигнала на быстрые и медленные колебательные функции. В результате разложения исходный сигнал может быть представлен в виде суммы амплитудно- и частотно-модулированных функций, называемых эмпирическими модами (ЭМ). Аналитическое выражение ДЭМ выглядит следующим образом:

$$x(n) = \sum_{i=1}^I IMF_i(n) + r_i(n),$$

где $x(n)$ – исходный сигнал; $IMF_i(n)$ – ЭМ; $r_i(n)$ – конечный остаток; $i = 1, 2, \dots, I$ – номер ЭМ; n – дискретный отсчет времени ($0 < n \leq N$, N – количество дискретных отсчетов в сигнале).

Среди всех разновидностей методов декомпозиции наиболее адаптивной к речевым сигналам является усовершенствованная полная множественная декомпозиция на эмпирические моды с введением адаптивного шума (ПМДЭМАШ) [8]. Применение улучшенной ПМДЭМАШ обеспечивает:

- адаптивное разложение, так как базисные функции, используемые при декомпозиции, извлекаются непосредственно из исходного речевого сигнала и позволяют учитывать только ему свойственные особенности (скрытые модуляции, области концентрации энергии и т.п.) [7];
- минимальный уровень остаточного шума и отсутствие паразитных ЭМ, возникающих на ранних этапах декомпозиции вследствие перекрытия масштабно-энергетических пространств мод [8].

Рассмотрим особенности предложенного способа повышения эффективности.

Форматный анализ (ФА). В спектре каждого звука речи присутствуют области повышенной концентрации энергии, называемые формантами. Анализ формант для задачи оценки психоэмоционального состояния заключается в определении числовых значений частоты и ширины полосы формантных областей, отражающих смещение частотной локализации формантных резонаторов вследствие патологии речеобразующей системы [9].

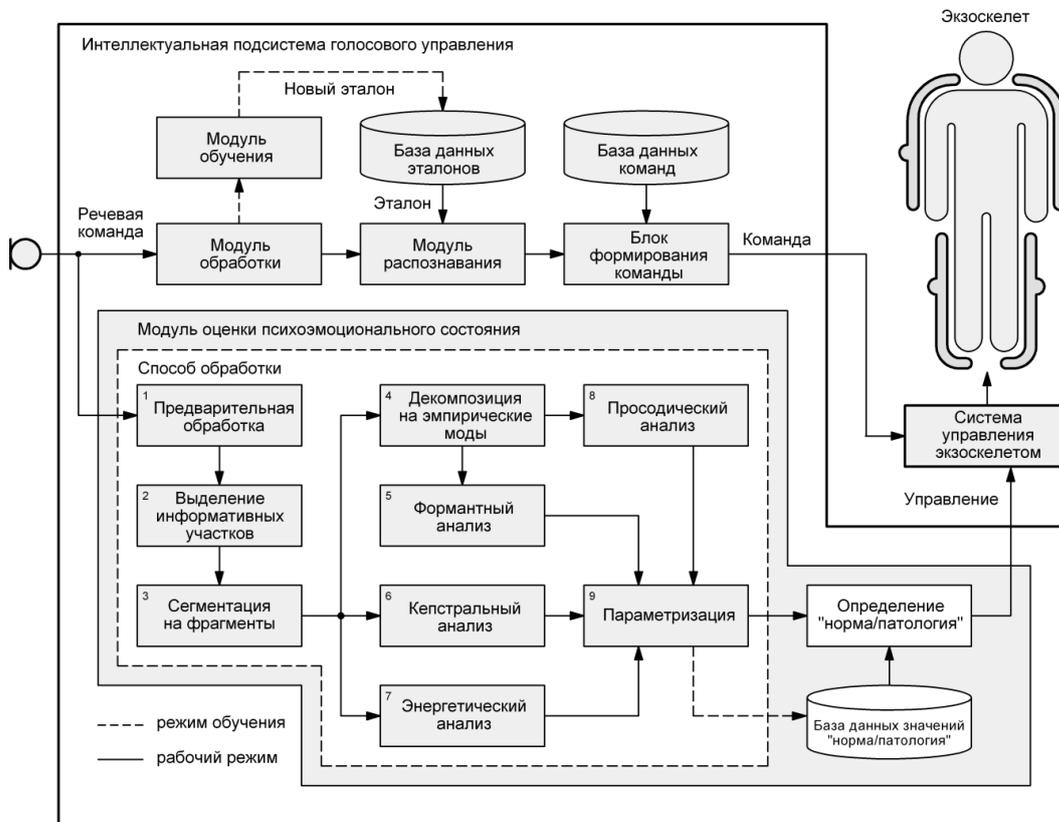


Рис. 1. Блок-схема способа обработки речевых команд в составе модуля оценки психоэмоционального состояния интеллектуальной подсистемы голосового управления медицинским экзоскелетом

Кепстральный анализ (КА). Для задачи оценки психоэмоционального состояния кепстральный анализ сводится к вычислению кепстральных коэффициентов на мел-шкале, называемых мел-частотными кепстральными коэффициентами (МЧКК) [10]. Особенностью кепстрального анализа является возможность исследования частотной характеристики речеобразующей системы человека, игнорируя характеристики сигналов возбуждения.

Энергетический анализ (ЭА). Амплитудное распределение энергии является важным информативным параметром психоэмоциональной составляющей речи [11]. Изменения амплитуды хорошо описываются с помощью функции кратковременной энергии сигнала. Приближая работу способа к функционалу слухового аппарата, для сжатия амплитуды сигнала в большом динамическом диапазоне применяется операция логарифмирования энергии.

Просодический анализ (ПА). Эффективное детектирование высокого и низкого уровня психоэмоционального состояния обеспечивают просодические характеристики речи, не являющиеся фонематическими и характеризующие речевую мелодию, темпоральные и тембральные особенности речи, её ритм, словесные тоны, стыки, паузы и интонации (т.е. фонацию основного тона на уровне фраз) [12]. Наиболее распространенные просодические характеристики речи: максимальное и среднее значения частоты основного тона (ЧОТ); стандартное отклонение контура ЧОТ; диапазон фонационных частот; джиттер; среднее абсолютное значение джиттера.

Для проведения исследований способа при поддержке центра психологической диагностики федерального казенного учреждения здравоохранения «Медицинская санитарная часть Министерства внутренних дел Российской Федерации по Пензенской области» были сформированы две группы испытуемых по 53 человека (мужского и женского пола, в возрасте от 18 до 79 лет) условно здоровых и с явно выраженной симптоматикой психоэмоциональных расстройств. Задачей исследования являлось обнаружение и оценка психоэмоционального расстройства. Для оценки эффективности использовался параметр – ошибки первого и второго рода. Все этапы обработки сигналов и анализа данных были выполнены в среде математического моделирования © Matlab (MathWorks). В табл. 1 представлены результаты оценки психоэмоционального состояния.

Таблица 1

Результаты оценки психоэмоционального состояния

| Прогнозируемый результат | Результат обнаружения | | Ошибки первого и второго рода, % | |
|--------------------------|-----------------------|---------|----------------------------------|------|
| | Патология | Норма | | |
| ЭА | | | | |
| Патология | 29 чел. | 24 чел. | 1-ого | 45,3 |
| Норма | 12 чел. | 41 чел. | 2-ого | 22,6 |
| ЭА + КА | | | | |
| Патология | 34 чел. | 19 чел. | 1-ого | 35,8 |
| Норма | 10 чел. | 43 чел. | 2-ого | 18,8 |
| ЭА + КА + ФА | | | | |
| Патология | 41 чел. | 12 чел. | 1-ого | 22,6 |
| Норма | 7 чел. | 46 чел. | 2-ого | 13,2 |
| ЭА + КА + ФА + ПА | | | | |
| Патология | 49 чел. | 4 чел. | 1-ого | 8,16 |
| Норма | 3 чел. | 50 чел. | 2-ого | 5,66 |

Как видно из результатов наименьшая ошибка первого рода 8,16 % достигается при использовании четырех видов анализа речевых команд в совокупности. Это объясняется тем, что каждая группа информативных параметров, предназначенная для описания отдельных аспектов речи релевантна психоэмоциональной составляющей. Четыре ложных присваиваний статуса «норма» речевым командам, произнесенным людьми с психоэмоциональными расстройствами, это допустимый показатель, меньший, чем порог (6 человек из 53), который условно определили авторы статьи в качестве удовлетворительного для корректной оценки психоэмоционального состояния операторов медицинских экзоскелетов. То же самое можно сказать про ложные присваивания статуса «патология» речевым командам, произнесенным условно здоровыми испытуемыми – 5,66 %.

Следующими этапами данной работы являются: определение оптимального количества информативных параметров, обеспечивающих корректную оценку психоэмоционального состояния операторов медицинских экзоскелетов; исследование способа на группах, состоящих из большого количества испытуемых.

Вывод: предложенный способ обработки речевых команд, повышающий эффективность оценки психоэмоционального состояния, может успешно тестироваться операторами медицинских экзоскелетов.

Библиографический список

1. Здравоохранение в России 2017 : стат. сб. – URL: http://www.gks.ru/free_doc/doc_2017/zdrav17.pdf (дата обращения: 21.02.2018).
2. Яцун, С. Ф. Экзоскелеты: анализ конструкций, классификации, принципы создания, основы моделирования : монография / С. Ф. Яцун, В. Е. Павловский, Б. В. Лушников, О. В. Емельянова. – Курск : Юго-Западный гос. ун-т, 2014. – С. 148.
3. Алимуратов, А. К. Особенности построения систем голосового управления для медицинских экзоскелетов расширенного функционала / А. К. Алимуратов, А. Ю. Тычков, И. И. Артёмов, М. Ю. Куликов // Качество. Инновации. Образование. – 2016. – № S2 (129). – С. 66–72.
4. Alimuradov, A. K. A method to determine cepstral markers of speech signals under psychogenic disorders / A. K. Alimuradov, A. Yu. Tychkov, A. V. Ageykin, P. P. Churakov // 2018 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBREIT) (May 7–8, 2018). – Yekaterinburg, Russia, 2017. – P. 128–131.
5. Алимуратов А. К. Обзор и классификация методов обработки речевых сигналов в системах распознавания речи / А. К. Алимуратов, П. П. Чураков // Измерение. Мониторинг. Управление. Контроль. – 2015. – № 2 (12). – С. 27–35.
6. Алимуратов, А. К. Оценка психоэмоционального состояния на основе анализа речевых сигналов: современное состояние, проблемы и перспективы / А. К. Алимуратов, А. Ю. Тычков, П. П. Чураков, А. В. Агейкин // Перспективные информационные технологии (ПИТ 2018) : тр. Международ. науч.-техн. конф. / под ред. С. А. Прохорова. – Самара : Изд-во Самар. науч. центра РАН, 2018. – С. 677–681.
7. Huang, N. E. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis / N. E. Huang, Sh. Zheng, R. L. Steven // Proc. R. Soc. Lond. – 1998. – A 454. – P. 903–995.
8. Vocal and facial biomarkers of depression based on motor incoordination and timing / J. R. Williamson, T. F. Quatieri, B. S. Helfer, G. Ciccarelli, D. D. Mehta // 4th International Workshop on Audio / Visual Emotional Challenge (AVEC). – 2014. – P. 65–72. – URL: <http://scholar.google.ru>
9. Фланаган, Дж. Л. Анализ, синтез и восприятие речи : пер. с англ. под ред. А. А. Пирогова / Дж. Л. Фланаган. – М. : Связь, 1968. – С. 395.
10. Рабинер, Л. Р. Цифровая обработка речевых сигналов : пер. с англ. / Л. Р. Рабинер, Р. В. Шафер. – М. : Радио и связь, 1981. – С. 496.
11. Huang, X. Spoken Language Processing / X. Huang, A. Acero, H.-W. Hon // Guide to Algorithms and System Development. Prentice Hall, Upper Saddle River. – N.J., 2001. – P. 980.
12. Schuller, B. W. Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing / B. W. Schuller, A. M. Batliner. – New York : Wiley, 2013. – P. 344.

С. Е. Вятчанин¹, Е. А. Малыгина¹, С. В. Туреев²

¹ Пензенский государственный университет, Пенза, Россия

² НИИСУС, Москва, Россия

ОСОБЕННОСТИ АВТОМАТИЧЕСКОГО ОБУЧЕНИЯ СЕТИ НЕЙРОНОВ «КРАМЕРА–ФОН МИЗЕСА» НА МАЛЫХ ВЫБОРКАХ

Показано, что алгоритм обучения больших сетей «Крамера–фон Мизеса» имеет линейную вычислительную сложность так же, как и алгоритм ГОСТ Р 52633.5–2011, что делает их перспективными при разработке новых приложений для систем биометрической аутентификации личности.

Нейронных сетей может быть много, так же, как и типов нейронов из которых они собраны. Для нейросетевой биометрии принципиально важным является то, чтобы нейронная сеть преобразователя биометрия-код была способна быстро обучаться на малой выборке примеров «Свой» в полностью автоматическом режиме. В частности, этими свойствами обладает сеть персептронов, обученных алгоритмом ГОСТ Р 52633.5 [1].

В соответствии с требованиями алгоритма ГОСТ Р 52633.5 [1] нейроны подключены ко входам нейросети случайным образом. При синтезе нейронной сети, связи каждого нейрона задаются таблицей, заполняемой от генератора псевдослучайных чисел. Для сети нейронов «Крамера-фон Мизеса» (KfM) связи каждого нейрона должны формироваться также [2].

Важнейшим аспектом автоматического обучения сетей KfM является вычисление вектора математических ожиданий $\bar{E}(v_i)$ и вектора стандартных отклонений $\bar{\sigma}(v_i)$ параметров 20-ти примеров образа «Свой». Эти данные запоминаются и используются для центрирования и нормирования (3) всех входных параметров нейронной сети [3].

Вторым важнейшим аспектом является вычисление средних значений вероятности для упорядоченной выборки биометрических параметров образа «Свой» $E(P(\tilde{v}_1))$, $E(P(\tilde{v}_2))$, ..., $E(P(\tilde{v}_n))$. Эти параметры далее используются как эталоны для сравнения с предъявленными нейрону данными:

$$\begin{cases} KfM(v) = \sum_{i=1}^n \{3(E(P(\tilde{v}_i) - P(\tilde{v}_i)))\}^2 \\ KfM(\xi) = \sum_{i=1}^n \{3(E(P(\tilde{v}_i) - P(\tilde{\xi}_i)))\}^2 \end{cases} \quad (1)$$

Система (1) содержит два уравнения, так как для данных образа «Свой» и для данных образа «Чужой» нейрон работает совершенно по-разному. Для данных образа «Свой», каждое из которых относительно мало происходит их квадратичное сжатие и накопление сумматором.

Для данных образа «Чужой» происходит не только линейное обогащение данных за счет суммирования, но и его нелинейное усиление за счет того, что нормировка биометрических данных «Чужой» выполняется на базе статистических данных «Свой»:

$$\tilde{\xi}_i = \frac{\xi_i - E(v_i)}{\sigma(v_i)} \quad (2)$$

Именно по этой причине с вероятностью выше 0.5 данные образа «Чужой» оказываются больше 1.0 и усиливаются квадратичной функцией. В конечном итоге выполняется условие:

$$KfM(\xi) > KfM(v) \quad (3)$$

Так как нейроны «Крамера-фон Мизеса» работают с вероятностями, на каждом входе сумматора таких нейронов должен стоять функциональный преобразователь континуумов входных состояний в вероятности их появления. По такой схеме каждый нейрон отвечает за один или два выходных разрядов кода аутентификации.

Принципиально важным свойством вероятностных нейронов KfM является очень высокая устойчивость их обучения на малых выборках.

Для того чтобы выполнить нормирование и центрирование биометрических данных под БиоОбраз «Свой» достаточно базы из 20 примеров. Как показала практика на базе обучающей выборки из 20 примеров образа «Свой» можно с достаточной точностью вычислить вектор математических ожиданий $\bar{E}(v_i)$ и вектор стандартных отклонений $\bar{\sigma}(v_i)$ всех контролируемых биометрических параметров.

Единственной не тривиальной функцией автомата обучения является вычисление по 20 примерам вектора математических ожиданий вероятности появления сортировки по возрастанию значений контролируемых биометрических параметров [4, 5].

При обучении нет переборных и запоминания промежуточных данных. Это означает, что обучение больших сетей искусственных нейронов «Крамера-фон Мизеса» имеет ли-

нейную вычислительную сложность, так же, как и алгоритм ГОСТ Р 52633.5 [1], что делает их перспективными при разработке новых приложений для систем биометрической аутентификации личности.

Библиографический список

1. ГОСТ Р 52633.5–2011. Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей «биометрия–код доступа».
2. Иванов, А. И. Сравнение мощности критерия среднего геометрического и Крамера–фон Мизеса на малых выборках биометрических данных / А. И. Иванов, Е. А. Малыгина, П. А. Перфилов, С. Е. Вятчанин // Модели, системы, сети в экономике, технике, природе и обществе. – 2016. – № 2. – С. 155–158.
3. Волчихин, В. И. Перспективы использования искусственных нейронных сетей с многоуровневыми квантователями в технологии биометрико-нейросетевой аутентификации / В. И. Волчихин, А. И. Иванов, В. А. Фунтиков, Е. А. Малыгина // Известия высших учебных заведений. Поволжский регион. Технические науки. – Пенза : Изд-во ПГУ, 2013. – № 4(28). – С. 88–99.
4. Иванов, А. И. Подавление шумов квантования биометрических данных при использовании многомерного критерия Крамера–фон Мизеса / А. И. Иванов, А. И. Газин, С. Е. Вятчанин // Проблемы информационной безопасности. Компьютерные системы. – 2016. – № 2. – С. 21–28.
5. Алгоритмы обучения и тестирования нейросетевых преобразователей «биометрия–код» : монография / Б. С. Ахметов, В. И. Волчихин, А. И. Иванов, В. А. Фунтиков, Е. А. Малыгина. – Алматы : Изд-во КазНТУ, 2014. – 136 с.

А. И. Иванов¹, Е. А. Малыгина², А. И. Солопов³

¹ Пензенский научно-исследовательский электротехнический институт, Пенза, Россия

² Пензенский государственный университет, Пенза, Россия

³ «18 ЦНИИ» МО РФ, Москва, Россия

ИСПОЛЬЗОВАНИЕ НЕЙРОНОВ КВАДРАТА СРЕДНЕГО ГЕОМЕТРИЧЕСКОГО ПЛОТНОСТЕЙ РАСПРЕДЕЛЕНИЯ ЗНАЧЕНИЙ ВЕРОЯТНОСТЕЙ ПРИ МНОГОМЕРНОМ АНАЛИЗЕ ДАННЫХ

Рассматриваются преимущества использования нейронов квадрата среднего геометрического плотностей распределения значений вероятностей при многомерном статистическом анализе данных.

При практической реализации многомерного статистического анализа очень удобным оказалось применение искусственных нейронных сетей [1], обучаемых стандартным алгоритмом [2] с линейной вычислительной сложностью и тестируемых после обучения стандартными алгоритмами [3]. Быстрое и абсолютно устойчивое автоматическое обучение может быть организовано для нейронных сетей, воспроизводящих хорошо исследованные радиально-базисные функции [3] или множество иных, менее изученных, квадратичных функционалов [4–9].

Практически любому известному статистическому критерию можно поставить в соответствие некоторый нейрон [10]. Их отличие будет состоять только в том, что нейрон требует обучения (настройки) тогда как статистические критерии, как правило, не настраивают (не регулируют) в части предобработки данных. При использовании статистических критериев необходима настройка только порогового элемента (необходимо выбрать значение требуемого показателя доверительной вероятности).

Так же как все квадратичные функционалы нейрон среднего геометрического со структурой, изображенной на рис. 1, всегда имеет положительный отклик линейной части.

Его настройка сводится к нормированию и центрированию m входных биометрических параметров по формуле:

$$u_i = \frac{E(v_i) - v_i}{\sigma(v_i)}, \quad (1)$$

где i – упорядоченные номера входов нейрона; $i = 1, 2, \dots, m$, связанные с 416, контролируемыми биометрическими параметрами БиоОбраза, например, полученного в среде моделирования «БиоНейроАвтограф» [9], таблица связей формируется заранее с использованием генератора псевдослучайных чисел, как это рекомендует стандарт [10].

После нормирования и центрирования (1) для выборки m параметров по n примерам образа «Свой» вычисляют математическое ожидание – $\bar{E}(u)$ и стандартное отклонение – $\bar{\sigma}(u)$ для нормального теоретического распределения – $\tilde{p}(u)$.

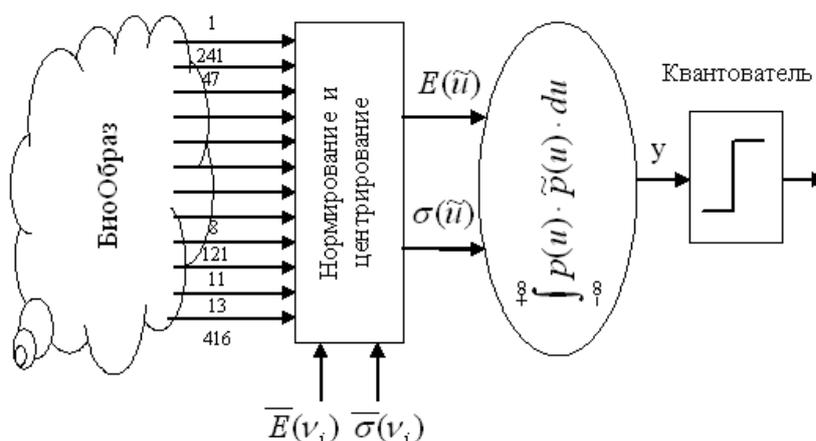


Рис. 1. Структурная схема искусственного нейрона, построенного как эквивалент квадрата среднего геометрического сравниваемых плотностей распределения значений

Если после настройки нейрона dsg^2 подать на его входы тестовые примеры образа «Свой», не участвовавшие в его обучении, то на выходе линейной части получим отклики с малой дисперсией:

$$y = \int_{-\infty}^{+\infty} (\tilde{p}(u))^2 \cdot du = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \left\{ \exp\left(\frac{-u^2}{2}\right) \right\}^2 du, \quad (2)$$

где математическое ожидание $E(u) \approx 0$ – практически не является случайной величиной, стандартное отклонение $\sigma(u) \approx 1$, так же практически не является случайной величиной.

Если же на входы обученного нейрона dsg^2 подавать биометрические данные образа «Чужой», то для них нормировка (1) работать не будет:

$$\tilde{u}_i = \frac{E(v_i) - \xi_i}{\sigma(v_i)}, \quad (2)$$

где ξ_i – биометрический параметр образа «Чужой».

Как следствие, математическое ожидание $E(\tilde{u})$ оказывается случайно величиной, а стандартное отклонение $\sigma(\tilde{u})$ принимает большие значения в интервале от 2 до 5. В итоге отклик – \tilde{y} нейрона dsg^2 на воздействие вектором биометрических параметров образа «Чужой» – $\bar{\xi}$ будет описываться уравнением совершенно не похожим на уравнение (2):

$$\tilde{y} = \int_{-\infty}^{+\infty} p(u) \cdot \tilde{p}(u) \cdot du = \frac{1}{2\pi \cdot \sigma(\tilde{u})} \int_{-\infty}^{+\infty} \left\{ \exp\left(\frac{-u^2}{2}\right) \right\} \cdot \left\{ \exp\left(-\frac{(E(\tilde{u}) - u)^2}{2 \cdot (\sigma(\tilde{u}))^2}\right) \right\} du, \quad (3)$$

где математическое ожидание $E(\tilde{u})$ – случайная величина с нулевым математическим ожиданием $E(E(\tilde{u})) \approx 0,0$ и значительным стандартным отклонением $\sigma(E(\tilde{u})) \approx 1,41$, стандартное отклонение $\sigma(\tilde{u}) \approx 3,8$ самой переменной не случайно и имеет значительную величину.

Кардинальное отличие уравнений состоит в том, что они дают совершенно разные по своей природе отклики. Уравнение (2) является почти детерминированным, тогда как уравнение (3) дает случайную величину с большим стандартным отклонением $-\sigma(\tilde{y})$. Именно это обстоятельство и давало возможность добиваться высокого уровня подавления шумов квантования, возникающих на малых выборках при применении критерия среднего геометрического от двух сравниваемых плотностей распределения значений [12]. Соотношения математических ожиданий распределения математических ожиданий образов «Свой», «Чужой» и их стандартных отклонений приведено на рис. 3 для нейронов dsg^2 с 8 входами (данные среды моделирования «БиоНейроАвтограф» [9]).

В силу того, что выражение (2) дает большое и почти детерминированное значение, а выражение (3) дает малое и случайное значение примеры образа «Свой» и примеры образов «Чужой» оказываются хорошо различимы, если использовать нейрон dsg^2 с 8 входами.

Результат разделения (одинаковая вероятность ошибок первого и второго рода) оказывается намного лучше, чем для линейного нейрона и обычного квадратичного нейрона: линейный нейрон с 8 входами $P_1 = P_2 = P_{EE} = 0,45$; квадратичный нейрон, имеющий 8 входов, $P_{EE} = 0,26$; нейрон dsg^2 с 8 входами $P_1 = P_2 = P_{EE} = 0,21$.

Таким образом, можно сделать вывод, что с ростом размерности нейросетевого преобразования выигрыш от замены линейных и квадратичных нейронов на нейроны среднего геометрического квадрата плотностей распределения значений вероятностей усиливается.

Библиографический список

1. ГОСТ Р 52633.0–2006. Защита информации. Техника защиты информации. Требования к средствам высоконадежной биометрической аутентификации. – М. : Стандартинформ, 2007. – 27 с.
2. ГОСТ Р 52633.5–2011. Защита информации. Техника защиты информации. Автоматическое обучение нейросетевых преобразователей «биометрия–код доступа». – М. : Стандартинформ, 2012. – 16 с.
3. Саймон, Х. Нейронные сети: полный курс / Х. Саймон. – М. : Вильямс, 2006. – 1104 с.
4. Ахметов, Б. Б. Многомерный статистический анализ биометрических данных сетью частных критериев Пирсона / Б. Б. Ахметов, А. И. Иванов, А. В. Безяев, Ю. В. Фунтикова // Алматы: Вестник Национальной академии наук Республики Казахстан. – 2015. – № 1. – С. 5–11.
5. Иванов, А. И. Подавление шумов квантования биометрических данных при использовании многомерного критерия Крамера–фон Мизеса / А. И. Иванов, А. И. Газин, С. Е. Вятчанин // Проблемы информационной безопасности. Компьютерные системы. – 2016. – № 2. – С. 21–28.
6. Ахметов, Б. Многомерные статистики существенно зависимых биометрических данных, порождаемые нейросетевыми эмуляторами квадратичных форм : монография / Б. Ахметов, А. Иванов. – Алматы : ЛЕМ, 2016. – 86 с.
7. Иванов, А. И. Снижение требований к размеру тестовой выборки биометрических данных при переходе к использованию многомерных корреляционных функционалов Байеса / А. И. Иванов, П. С. Ложников, А. Е. Сулавко // Инфокоммуникационные технологии. – 2017. – № 15 (2). – С. 186–193.
8. Иванов, А. И. Идентификация подлинности рукописных автографов сетями Байеса–Хэмминга и сетями квадратичных форм / А. И. Иванов, П. С. Ложников, Е. И. Качайкин // Вопросы защиты информации. – 2015. – № 2. – С. 28–34.
9. Иванов, А. И. Среда моделирования «БиоНейроАвтограф» (программный продукт создан лабораторией биометрических и нейросетевых технологий) / А. И. Иванов, О. С. Захаров. – URL: <http://пниэи.рф/activity/science/noc.htm> (дата обращения: 10.08.2018).
10. Иванов, А. И. Многомерная нейросетевая обработка биометрических данных с программным воспроизведением эффектов квантовой суперпозиции : монография / А. И. Иванов. – Пенза : Изд-во АО «ПНИЭИ», 2016. – 133 с.
11. Иванов, А. И. Оценка соотношения мощностей семейства статистических критериев «среднего геометрического» на малых выборках биометрических данных / А. И. Иванов, К. А. Перфилов // Современные охраняемые технологии и средства обеспечения комплексной безопасности объектов : материалы XI Всерос. науч.-практ. конф. – Пенза ; Заречный, 2016. – С. 223–229.
12. Иванов, А. И. Оценка качества малых выборок биометрических данных с использованием дифференциального варианта статистического критерия среднего геометрического / А. И. Иванов, К. А. Перфилов, Е. А. Малыгина // Вестник СИБГАУ. – 2016. – № 4 (17). – С. 864–871.

А. С. Башев

Пензенский государственный университет, Пенза, Россия

МЕТОД ДИНАМИЧЕСКОГО ИЗМЕНЕНИЯ ПАРАМЕТРА БДИТЕЛЬНОСТИ В НЕЙРОННОЙ СЕТИ ART-1

Кластерный анализ является важной частью работы с большими объемами данных, позволяя достичь нескольких целей: сократить избыточную выборку, используя только наиболее типичных представителей каждого кластера; выявить элементы выборки, не относящиеся ни к одному из кластеров; использовать для каждого кластера данных свой метод анализа; создать таксономию [1] данных, используя иерархическую кластеризацию.

Алгоритм кластеризации – это функция $a: X \rightarrow Y$, которая любому объекту $x \in X$ ставит в соответствие номер кластера $y \in Y$. Множество Y в некоторых случаях известно заранее, однако чаще ставится задача определить оптимальное число кластеров, с точки зрения того или иного критерия качества кластеризации. Задача кластеризации и отличается от задачи классификации объектов тем, что метки исходных объектов Y_i изначально не заданы, и даже может быть неизвестно само множество Y (в таком случае задача кластеризации относится к широкому классу задач обучения без учителя)

Решение задачи кластеризации принципиально неоднозначно, и тому есть несколько причин:

- Не существует однозначно наилучшего критерия качества кластеризации. Известен целый ряд эвристических критериев, а также ряд алгоритмов, не имеющих чётко выраженного критерия, но осуществляющих достаточно разумную кластеризацию «по построению». Все они могут давать разные результаты.
- Число кластеров, как правило, неизвестно заранее и устанавливается в соответствии с некоторым субъективным критерием.
- Результат кластеризации существенно зависит от метрики, выбор которой, как правило, также субъективен и определяется экспертом.

В рамках данной работы условимся, что успешным результатом кластеризации будет считаться тот, в котором полученное количество кластеров эталонным количеством кластеров выборки. Ошибкой кластеризации будем называть несовпадение полученного кластера объекта с эталонным.

Существует множество методов решения задачи кластеризации данных из которых выделим нейросетевой подход.

Нейронные сети относятся к классу нелинейных адаптивных систем с архитектурой, условно имитирующей нервную ткань, состоящую из нейронов. Среди нескольких моделей нейронных сетей используемых для решения задач кластеризации, в рамках данной работы рассмотрим способных обучаться без учителя, как более универсальных в плане работы с реальными данными.

Решим задачу кластеризации набора данных NIST [2] предлагающий изображения цифр от 1 до 10 разрешением 8 на 8 точек в оттенках серого, выбранный из-за своей пространственности и простоты. Количество объектов в данном наборе данных – 1797.

Условимся, что мы опустим знание об известном эталонном количестве кластеров (10) в процессе настройки сети, а признаковым описанием объектов выборки будет служить 64 атрибута – значения для каждой точки изображения объекта. Из-за данного условия, например, становится неприемлемым использовать, например, сеть Кохонена, поскольку её настройка требует значение количества кластеров в наборе данных.

Сеть адаптивного резонанса ART-1, напротив, способна увеличивать число нейронов, соответствующих эталонным образцам кластера в процессе функционирования сети, число таких нейронов (емкость сети) разумеется, конечно, но не обязано соответствовать количеству кластеров в исследуемом наборе данных. ART-1 может создать новые категории и отбросить входные примеры, в случае исчерпания своей емкости. Однако число обнаруженных сетью категорий чувствительно к параметру бдительности, задаваемому при конфигурации сети [3].

Модель нейронной сети ART-1 подходит для решения поставленной задачи, если бы не одно «но», к сожалению, установка параметра бдительности необходимого при конфигурации сети – это субъективный процесс, зависящий от мнения эксперта.

Установка большого значения параметра бдительности влечет за собой неконтролируемый рост количества нейронов образцов при работе сети ART-1 и увеличивает шанс запуска процедуры поиска. В ходе этой процедуры распознающие нейроны отключаются один за другим с помощью функции сброса пока порог бдительности не будет преодолен. На каждом цикле поиска выбирается наиболее активный распознающий нейрон, и отключается, в случае если активация не достигает порога бдительности, что при большом количестве емкости сети, замедляет её работу, а в худшем случае приводит к преждевременному исчерпанию емкости сети.

Установка малого значения параметра бдительности увеличивает количество ошибок кластеризации, поскольку объекты объединяются в слишком большие общие кластеры.

И в том и в ином случае уменьшается шанс того что результат кластеризации будет успешным, поскольку полученное количество кластеров чаще всего будет больше или меньше эталонного соответственно.

Данная проблема частично решается моделью нейронной сети ARTMAP.

ARTMAP состоит из двух модифицированных ART-1 сетей, вторая сеть способна влиять на назначение параметра бдительности первой сети в значимости от соответствия полученного кластера и эталонного из обучающей выборки [4]. Но данная модель теряет способность обучаться без учителя как ART-1 или ART-2 сети.

Так же существует подход создания ансамбля из n ART-1 сетей, где выполняется условие:

$$V_i < V_{i+1}, i = 1..n - 1,$$

где V – параметр бдительности сети

Результирующий кластер объекта задается как конкатенация кластера полученного в работе каждой последующей сети начиная с 1 до n , т.е. если использует ансамбль из 3 ART-1 сетей с $V_1 = 0.2, V_2 = 0.5, V_3 = 0.8$, а полученные кластеры для определенного объекта выборки, то результирующим кластером для данного объекта будет.

Но такой подход не уменьшает влияние установки параметра бдительности, а лишь позволяет более точно характеризовать кластер объекта и дает возможность просто представить результаты кластеризации в виде дендрограммы.

Предлагаемый метод динамического изменения параметра бдительности в нейронной сети ART-1 подразумевает построение регрессионного дерева в листах которого находятся значения функции отклонения параметра бдительности. Он основан на идее о том, что каждая проверка схожести образца и объекта на входном слое нейронной сети с помощью параметра бдительности может быть представлена как узел двоичного дерева, где левое поддерево, представляет успешную проверку схожести, а правое поддерево – сброс нейрона победителя и запуск процедуры поиска.

При конфигурации сети, установим следующие значения $V = 0, j = 0$ (указатель на корень дерева).

Далее, для каждого объекта выборки S_i , необходимо

1. Вычислить значение сброса R_i и индекс нейрона победителя W_i (кластера).
2. Если $V \leq 0, V = R_i$
3. Если $V > 1, V = R_i$
4. Если $j = 0$, создать корневой узел $T_j, R_i \leq V$

5. Создать левый дочерний узел T_j^l , удовлетворяющий условию $R_i \leq V$, записать туда $W_j^l = W_i, R_j^l = R_i$
6. Создать правый дочерний узел T_j^r , удовлетворяющий условию $R_i > V$, записать туда $W_j^r = W_i, R_j^r = R_i$
7. Создать левый дочерний узел T_j^{ll} , удовлетворяющий условию $W_j^l = W_i$, записать туда $D_j^{ll} = V^2 - (R_i)^2$
8. Создать правый дочерний узел T_j^{lr} , удовлетворяющий условию $W_j^l \neq W_i$, записать туда $D_j^{lr} = (R_i)^2 - V^2$
9. Создать левый дочерний узел T_j^{rl} , удовлетворяющий условию $W_j^r = W_i$, записать туда $D_j^{rl} = (R_i)^2 - V^2$
10. Создать правый дочерний узел T_j^{rr} , удовлетворяющий условию $W_j^r \neq W_i$, записать туда $D_j^{rr} = V^2 - (R_i)^2$
11. Произвести обход полученного дерева сверху-вниз до терминального листа.
12. $V = V + D$
13. Установить указатель j на терминальный лист.

На рис. 1 представлена работа алгоритма для первого объекта набора данных.

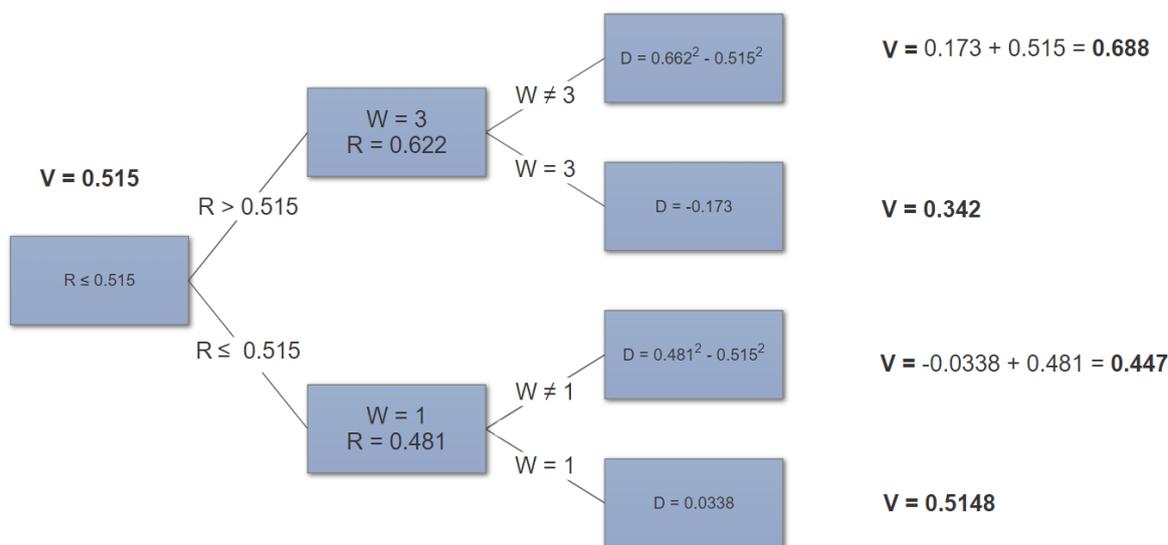


Рис. 1. Дерево коррекции значения параметра бдительности

Проведем эксперимент, подтверждающий состоятельность предложенного метода, по сравнению с эталонной ART-1 сетью с субъективно установленным значением параметра бдительности при решении поставленной задачи.

Поскольку сеть ART-1 работает только с двоичными входными значениями необходимо преобразовать изображения исходного набора данных с оттенков серого к черно-белому представлению и сгенерировать несколько наборов, в которых объекты перемешены случайным образом, в рамках данной работы выбрана цифра 100. Эталонная сеть ART-1 и сеть использующая предложенный метод динамического изменения параметра бдительности будут пересоздаваться для каждого из полученных наборов данных при решении задачи его кластеризации.

Результаты проведения эксперимента приведены в табл. 1.

Результаты проведения кластеризации совокупности наборов данных NIST сетями ART-1

| Эталонная сеть | | Предложенный метод |
|----------------|---|--------------------|
| Значение | Описание | Значение |
| 64 | Размер слоя входных нейронов | 64 |
| 100 | Размер слоя нейронов образцов, емкость сети | 100 |
| 0.89 | Значение параметра бдительности | – |
| 0.8 | Значение параметра скорости обучаемости сети | 0.8 |
| 138369 | Общее количество успешных решений задачи кластеризации за все прогоны | 125790 |
| 48519 | Общее количество ошибок кластеризации за все прогоны | 64692 |

Предложенный метод позволяет сети ART-1 работать без предварительного подбора параметра бдительности и может быть использован в других моделях нейронных сетей адаптивного резонанса (таких как ART-2).

Как видно из результатов проведенного эксперимента недостатком предложенного метода является падения качества результата решения задачи кластеризации по сравнению с субъективной установкой значения параметра бдительности, но полученные отклонения находятся в рамках нормы других алгоритмов кластеризации с неопределенным количеством кластеров таких как XMEANS. Дальнейшая работа в этом направлении будет посвящена улучшениям предложенного метода необходимых для приближения к эталонным результатам.

Библиографический список

1. Таксономия (теория классификации) // Струнино – Тихорецк. – М. : Советская энциклопедия, 1976.– (Большая советская энциклопедия : [в 30 т.] / гл. ред. А. М. Прохоров. – М., 1969–1978. – Т. 25).
2. NIST Form-Based Handprint Recognition System / M. D. Garris, J. L. Blue, G. T. Candela, D. L. Dimmick, J. Geist, P. J. Grother, S. A. Janet, C. L. Wilson // NISTIR, 1994. – 5469. –URL: <http://archive.org>
3. Carpenter, G. A. The Handbook of Brain Theory and Neural Networks, Second Edition / G. A. Carpenter, S. Grossberg. – Cambridge, MA : MIT Press, 2003. – P. 87–90.
4. Carpenter, G. A. Neural Networks / G. A. Carpenter, S. Grossberg, J. H. Reynolds // ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. – 1991. – № 4. – P. 565–588.

О. Н. Бодин, Г. Ф. Убиенных, А. Г. Убиенных

Пензенский государственный университет, Пенза, Россия

МОДЕЛИРОВАНИЕ ВИХРЕВЫХ АВТОВОЛН ТИПА РИЕНТРИ В ПРЕДСЕРДИЯХ ЧЕЛОВЕКА

Проанализированы основные механизмы возникновения сердечных аритмий с точки зрения их пространственного размера. На основе уравнений ФицХью – Нагумо в среде MATLAB проведено концептуальное моделирование аритмий, обусловленных механизмом риентри. Полученные результаты позволят повысить эффективность диагностики сердечных аритмий у пациентов.

Сердечные аритмии являются важной проблемой здравоохранения, так как в значительной степени связаны с повышенным риском сердечно-сосудистых осложнений и вне-

запной смерти, приводят к снижению качества жизни, инвалидности, высокой смертности, и повышению экономических затрат здравоохранения. В связи с этим изучение распространенности нарушений ритма и проводимости сердца является одной из актуальных задач современной кардиологии. Строго говоря, сердечная аритмия – это любое отклонение сердцебиения от строгой периодичности. В этом смысле все живые существа все время страдают аритмией, поскольку нормальное сердцебиение никогда не бывает точно периодическим. Однако представляющими интерес аритмиями являются те, которые считаются ненормальными и которые имеют нетривиальные физиологические последствия.

Прежде чем обсуждать сердечные аритмии, дадим описание нормального сердечного ритма. Человеческое сердцебиение – это сокращение сердечной мышцы, которое стимулируется электрическим событием, происходящим примерно один раз в секунду, в течение которого каждая клетка в сердце испытывает быстрые изменения трансмембранного потенциала, продолжающиеся около 300 мс, прежде чем вернуться в состояние покоя. Эти изменения, называемые потенциалом действия, являются волной электрической активности. Эта волна электрической активности инициируется небольшим скоплением клеток в правом предсердии около верхней полой вены, называемом синоатриальным (синусным, синусно-предсердным) узлом (SA). Эти клетки являются автономными генераторами и синхронизируются локальной электрической связью. После инициирования волна электрической активности распространяется по предсердиям, заканчиваясь на предсердной желудочковой перегородке (AV-перегородке), непроводящей перегородке между предсердиями и желудочками. Эта волна активности входит в еще один небольшой участок клеток в AV-перегородке, атриовентрикулярный (предсердно-желудочковый) узел (AV-узел), через который она медленно распространяется. Покидая AV-узел, волна входит в пучок специализированных волокон, называемых системой Гиса-Пуркинье, состоящей из пучка Гиса, ветвей связок и волокон Пуркинье, которые разветвляются древовидным образом, чтобы достичь внутренней (эндокардиальной) стенки желудочков. Возникает возбуждение у волокон Пуркинье для активации желудочковой ткани, распространяющееся изнутри наружу, заканчивая эпикардом.

Мышечные клетки сердца подразделяются на три типа, каждый из которых выполняет свои функции. Одни клетки находятся в колебательном режиме, другие возбуждаются и распространяют электрический потенциал. Наконец, остальные возбуждаются и под действием указанного потенциала сокращаются, способствуя этим выталкиванию крови. Каждая клетка специализирована на контакт с близлежащими клетками.

Активный потенциал, приводящий к сокращению предсердий и желудочков сердца, генерируется в специальных клетках, называемых пейсмекерами (водителями ритма), или P-клетками. Указанные клетки входят в состав всех отделов проводящей системы сердца. При этом наблюдается градиент автоматии сердца, т.е. снижение степени автоматизации различных участков сердца. У человека в покое частота возбудимости в 1 мин SA-узла – 60–80, AV-узла – 40–50, пучка Гиса – 30–40, волокон Пуркинье – около 20. В нормальных физиологических условиях генератором автоматии сердца является SA-узел, активность остальных очагов автоматии подавлена.

Профиль бегущей волны передающих клеток проводящей системы сердца (второй тип клеток) примерно симметричный, имеет амплитуду около 80 мВ, длительность приблизительно 0.2 с. Особое положение занимают волокна Пуркинье. Они содержат ансамбли как пейсмекерных клеток, так и передающих клеток.

Клетки третьего типа, составляющие миокард предсердий и желудочков, называемые кардиомиоцитами, имеют сократительный механизм, подобный поперечнополосатым (скелетным) мышцам. Механизм сокращения этих мышц сравнительно хорошо изучен.

Стандартным диагностическим инструментом для клинической оценки сердечного ритма является ЭКГ, которая измеряет потенциал поверхности тела, создаваемый движущейся волной. Нормальная ЭКГ имеет три характерных сигнала: P-волну, соответствующую активации предсердий, комплекс QRS, соответствующий активации желудочков, и T-волну, соответствующую восстановлению (возврат к покою) желудочков.

Диагностические критерии нормального синусового ритма [1]:

- положительный зубец Р во II стандартном отведении;
- постоянный и нормальный интервал PQ;
- постоянная форма зубца Р в каждом отведении;
- постоянное расстояние Р-Р или R-R.

Рассмотрим основные механизмы аритмий с точки зрения их размерности.

Аритмии нулевой размерности. Аритмиями нулевой размерности являются те, которые связаны с одиночными клетками или небольшими наборами клеток, и не полагаются на пространственное расположение клеток. Эти клетки изменяют свое поведение от нормального до определенного типа ритмического или аритмического поведения. Например, клетки SA-узла являются автономными осцилляторами, и нарушение автоматизма их работы вызывает нотопные аритмии (синусовая тахикардия, синусовая брадикардия, синусовая аритмия). При этом пейсмейкером остается SA-узел, но происходит изменение параметров следования импульсов. Другой пример – эктопический центр, представляющий собой совокупность клеток, которая подавляет возбудимость SA-узла и становится новым пейсмейкером. Эктопические центры могут появляться в предсердиях, AV-перегородке, желудочках сердца. Это приводит к гетеротопным аритмиям, задающим замедленные или ускоренные ритмы.

Одномерные аритмии. Одномерные аритмии названы так потому, что их существование опирается на одномерный путь распространения. Классический пример одномерной аритмии связан с синдромом преждевременного возбуждения Вольфа-Паркинсона-Уайта (Wolff-Parkinson-White). Для него характерна аномальная проводимость сердечной ткани между предсердиями и желудочками, проявляющаяся в возвратной тахикардии и суправентрикулярной тахикардии. Участок, являющийся причиной заболевания, называется пучком Кента. Он может быть расположен между двумя предсердиями и соединяется с одним любым желудочком. При таком синдроме потенциал действия поступает в желудочки через пучок Кента, распространяется ретроградно вдоль волокон Пуркинье к AV-узлу, отсюда в предсердия, вызывая циркуляцию волны. Постоянно циркулирующий волновой фронт приводит к очень быстрому сердечному ритму с высокой активностью накачки крови.

Одномерные реципрокные тахикардии были впервые изучены в начале 1900-х годов Майнсом (G. R. Mines) [2], когда он намеренно отрезал кольцо ткани вокруг верхней полой вены и сумел инициировать волны, которые проходили только в одном направлении.

Двумерные аритмии. Двумя наиболее распространенными реципрокными аритмиями являются тахикардия и фибрилляция. Обе они могут возникать в предсердиях (предсердная тахикардия и предсердная фибрилляция) или в желудочках (желудочковая тахикардия и фибрилляция желудочков). Когда они возникают в желудочках, они опасны для жизни.

Двумерные аритмии связаны с двумерными самоподдерживающимися волнами активности, спиральными волнами, такими как встречающиеся в предсердиях во время предсердной тахикардии или фибрилляции. Причина, по которой эти аритмии считаются дву-, а не трехмерными, заключается в том, что стенка предсердий довольно тонкая.

Трехмерные аритмии. Трехмерные аритмии вызываются самоподдерживающимися волнами электрической активности, которые происходят в желудочках, и называются свитковыми волнами или волнами прокрутки. Они трехмерные, а не двумерные, потому что желудочковая стенка довольно толстая. Предположение, что волны ориентри в желудочках должны быть подобны свитку, было сделано довольно рано в работе Уинфри (A.T. Winfree) [3], однако экспериментальная проверка этого наступила гораздо позже.

Наблюдение, что ориентеративные аритмии в желудочках являются трехмерными, имеет важные клинические последствия. Если аритмия одномерна, тогда должна быть точка где-то в цепи, которая может быть разрушена, что устранил аритмию. Аналогично, если двумерная аритмия является результатом быстрой периодической стимуляции от эктопического центра (т.е. неповторного входа), то разрушение осциллятора устранил аритмию. Однако, если точка на поверхности желудочков является точкой первой активации, нет никакой гарантии, что эта точка является эктопическим фокусом или точкой в одномерной схеме. Фактически, ориентеративная аритмия, циркулирующая глубоко внутри ткани, может

иметь точку на поверхности, на которой она впервые появляется, но поскольку аритмия поддерживается трехмерным повторным циклом, разрушение поверхностных клеток не будет иметь никакого влияния на аритмию.

Одной из причин возникновения аритмий является механизм повторного обратного входа волны, или риентри (от англ. re-entry). Риентри представляет собой режим, при котором волна возбуждения циркулирует по кругу. Такая циркуляция может возникнуть вокруг естественной неоднородности (шрама, крупной артерии), если каким-то образом симметрия проведения возбуждения оказывается нарушена, и циркуляция начинается в одном из направлений. К примеру, возникает участок, который пропускает волны возбуждения только в одном направлении. Такой эффект называется однонаправленным блоком и является необходимым условием возникновения риентри (рис. 1).

Существует множество моделей описания сердечной ткани, учитывающее перечисленные выше эффекты. Так или иначе, все модели состоят из двух частей: реакции в точке и распространения импульса в пространстве.

По типу описания среды, в которой распространяется волна, модели сердечной ткани могут быть разделены на три класса: монодоменные, бидоменные и мультидоменные. Монодоменные модели рассматривают возбудимую среду как единый резервуар, в котором ионы распространяются посредством диффузии. В бидоменных моделях внутренность клеток (клеточный синцитий) и межклеточное пространство рассматриваются как два отдельных резервуара с различными концентрациями ионов. Мультидоменные модели содержат множество резервуаров, которыми могут быть внутренние объемы отдельных клеток, межклеточные щели и т.д.

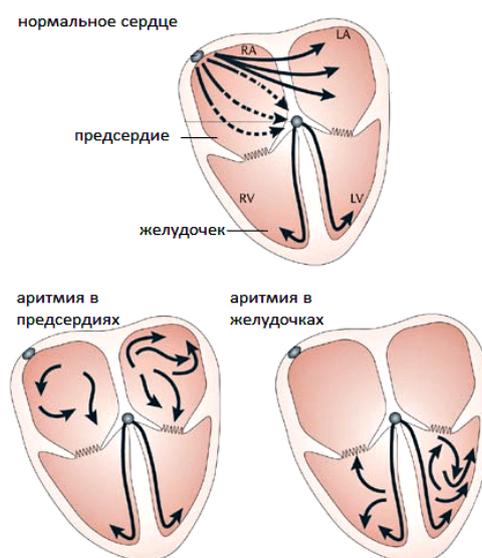


Рис. 1. Нормальное и anomальное распространение возбуждения по сердечной ткани

Монодоменные модели отличаются между собой компонентой, ответственной за реакцию. К числу монодоменных моделей с упрощенным реакционным членом относятся модели ФитцХью–Нагумо [4, 5] и Алиева-Панфилова [6], активно используемые ныне для качественного описания фундаментальных эффектов в сердечной ткани.

В настоящей работе использовались уравнения ФитцХью–Нагумо

$$\frac{\partial u}{\partial t} = C\varepsilon^{-1} \left(u - \frac{u^3}{3} - v \right) + \Delta u \quad (1)$$

$$\frac{\partial v}{\partial t} = \varepsilon (u + \beta - \gamma v)$$

с параметрами $\varepsilon = 0,03$, $\beta = 0,7$, $\gamma = 0,8$. Эти уравнения, как известно, являются наиболее подходящими для концептуальных исследований волн вихревого типа [7].

Уравнения (1) в двумерной области имеет бесконечно большое количество топологически различных решений. Конкретное решение из бесконечно большого их количества может быть получено путем задания специально подобранных начальных условий или специально подобранных внешних воздействий. В работе исследовались следующие способы возбуждения волн риентри:

1. Стимуляция по двум точкам, где одна точка располагается в центре области, а другая – на частично рефрактерной первой волны возбуждения.

2. Стимуляция «кросс-поле», где один стимул прикладывается к левой границе области, вызывая плоскую волну, а другой – к нижней границе области.

Моделирование проводилось в среде MATLAB 8 в прямоугольной области размером 128×128 , уравнения решались с помощью метода конечных разностей для пространственных производных и явным интегрированием Эйлера для производных по времени. Применялись граничные условия Неймана. Результаты моделирования представлены на рис. 2.

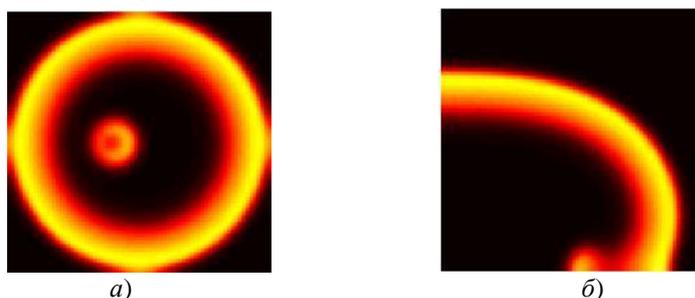


Рис. 2. Волна возбуждения:
а – стимуляция по двум точкам; б – стимуляция «кросс-поле»

Проведенные вычислительные эксперименты подтверждают возможность существования разнообразных топологически различных волн риентри.

Дальнейшая работа представляется в исследовании трехмерных аритмий с реалистической геометрией сердца человека.

Библиографический список

1. Кушаковский, М. С. Аритмии сердца / М. С. Кушаковский, Ю. Н. Гришкин. – СПб. : Фолиант, 2014. – 720 с.
2. Mines, G. R. On circulating excitations in heart muscle and their possible relation to tachycardia and fibrillation / G. R. Mines // Trans. Roy. Soc. Can. – 1914. – Vol. 4. – P. 43–59.
3. Winfree, A. T. Scroll-shaped waves of chemical activity in three dimension / A. T. Winfree // Science. – 1973. – Vol. 181. – P. 937–939.
4. FitzHugh, R. Impulses and physiological states in theoretical models of nerve membrane / R. FitzHugh // Biophysical J. – 1961. – Vol. 1, № 6. – P. 445–466.
5. Nagumo, J. An active pulse transmission line simulating nerve axon / J. Nagumo, S. Arimoto, S. Yoshizawa // Proc. IRE. – 1962. – № 50. – P. 2061–2070.
6. Aliev, R. R. A simple two-variable model of cardiac excitation / R. R. Aliev, A. V. Panfilov // Chaos Solutions and Fractals. – 1996. – Vol. 7, № 3. – P. 293–301.
7. Алиев, Р. Р. Концептуальные и детальные модели электрической активности миокарда : автореф. дис. ... д-ра физ.-мат. наук / Алиев Р. Р. – Пушкино, 2007. – 46 с.

А. С. Никонов, И. П. Бурукина

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ ДЛЯ РАСПОЗНАВАНИЯ РЕЧИ

Предложен подход к решению задачи распознавания речи. Описано применение вейвлет-преобразования

в рассматриваемой задаче. Рассмотрены некоторые аспекты данной тематики.

С самого появления человек наделяется органами чувств. Все они помогают ему получать информацию о внешнем мире и общаться с ним, в том числе и с другими людьми.

Благодаря своему активному развитию, вычислительные устройства стали незаменимым инструментом для человека. Современная жизнь наполнена взаимодействием с вычислительной техникой и программным обеспечением.

Создание удобного интерфейса для взаимодействия с человеком является неотъемлемой частью разработки ПО. Относительно недавно, некоторые устройства или программы обеспечиваются возможностью голосового управления. Создание такой функциональности обусловлено не только удобством, но и необходимостью. Она очень важна для людей, имеющих трудности с речевым или слуховым аппаратами.

Задача распознавания речи состоит в восстановлении по звуковому сигналу слова естественного языка (из ограниченного словаря), произнесением которого является этот звуковой сигнал [1]. Распознавание речи является задачей классификации образов акустических характеристик речевых сигналов [2]. Обратной является задача синтеза речи.

Основным подходом в решении данной задачи является создание словаря, который хранит некоторые эталонные значения. Следовательно, программа должна сравнить полученные значения с уже имеющимися эталонами в словаре и сделать вывод о том, что было произнесено.

Исходя из этого, системы распознавания речи можно классифицировать по степени детализации при задании эталонов [1]: системы, использующие в качестве эталонов слова и системы, использующие в качестве эталонов элементы слов.

При таком подходе возникает вопрос о размере словаря. Он определяется решаемой задачей.

В описываемом подходе предлагается в качестве элемента словаря использовать слова. Таким образом, входными данными являются слова, произнесенные диктором.

Для разработки компонентов, выполняющих распознавание речи, необходимо представлять природу звука и восприятие его человеком компьютером.

Понятие звука связано с понятием волны. Звук возникает в результате неких возмущений, которые вызывают изменение давления в некоторой точке пространства. Этот перепад давления передается между соседними слоями воздуха. Таким образом, звук распространяется в виде звуковой волны от источника к приемнику.

Органы слуха человека имеют сложную организацию. Основное назначение этих органов – преобразование звука в нервные импульсы.

В технике под звуковой волной понимается зависимость амплитуды от частоты. Звуковая волна – это сложная функция. Для ее представления в компьютере, выполняют дискретизацию.

В персональном компьютере для работы со звуком имеется звуковая карта. Она выполняет обработку сигнала, в том числе дискретизацию и квантование. В результате сигнал в виде последовательности двоичных данных передается в память. При воспроизведении звука выполняется обратная процедура (цифро-аналоговое преобразование).

Таким образом, на вход системы поступает последовательность байт, описывающих уровни звукового сигнала. Основным вопросом является использование этих данных при распознавании.

Вейвлет-преобразование является относительно новым математическим аппаратом. Основным понятием является «вейвлет». Это математическая функция в виде волны (всплеска), удобная для анализа сигналов. Этот термин был введен Хааром и впоследствии проработан Добеши и Морле.

Вейвлет-преобразование представляет собой скалярное произведение вейвлета и анализируемого сигнала [3]. Оно позволяет разделить сигнал на две составляющие: аппроксимирующую (низкочастотную) и детализирующую (высокочастотную).

Аппроксимирующая составляющая несет основную информацию о сглаженном сигнале, а детализирующая – детали и, возможно, шум.

В результате вейвлет-преобразования сигнал сжимается, но сохраняется его форма.

Многоуровневое вейвлет-преобразование связано с алгоритмом Маллата. Он заключается в многократном выполнении декомпозиции аппроксимирующей составляющей и отбрасывании детализирующей. При этом на первом шаге рассматривается сам сигнал.

Таким образом, основная идея подхода к распознаванию речи заключается в следующем. Произнесенное слово представляется в виде последовательности закодированных уровней сигнала. Для этого требуется разработка соответствующих компонентов. Приложение выполняет многоуровневое вейвлет-преобразование по алгоритму Маллата. В результате получается сглаженное, сжатое представление произнесенного слова.

Работа программы основывается на наличии словаря (набора поддерживаемых слов). Приложение должно поддерживать два основных режима работы: обучение и распознавание. При обучении система должна сохранять сформированные данные для каждого слова. При распознавании преобразованные данные следует сравнить с полученными ранее в процессе обучения.

Таким образом, заключительным этапом является нахождение соответствия между представлением произнесенного слова и содержимым словаря. Для этой цели возможно использование алгоритма DTW (dynamic time warping, динамической трансформации временной шкалы).

Алгоритм позволяет сравнивать последовательности разной длины и находить стоимость пути между двумя последовательностями. Решая эту задачу оптимизации, можно определить, какое слово наиболее близко к произнесенному.

Таким образом, идея распознавания слова основана на выполнении вейвлет-преобразования входного сигнала и дальнейшем решении задачи с использованием алгоритма динамической трансформации временной шкалы.

Библиографический список

1. Мазуренко, И. Л. Компьютерные системы распознавания речи / И. Л. Мазуренко // Интеллектуальные системы. – 1998. – № 1–2. – С. 117–133.
2. Запрягаев, С. А. Распознавание речевых сигналов / С. А. Запрягаев, А. Ю. Коновалов // Вестник ВГУ. Сер.: Системный анализ и информационные технологии. – 2009. – № 2. – С. 39–48.
3. Астафьева, Н. М. Вейвлет-анализ: основы теории и примеры применения / Н. М. Астафьева // Успехи физических наук. – 1996. – № 11. – С. 1145–1170.

А. А. Валько, А. Ф. Валько, А. С. Ивлиев

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ВОЗМОЖНОСТЕЙ СОВРЕМЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОРРЕКТИРОВКИ ВЛИЯНИЯ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ НА ПОЛЬЗОВАТЕЛЯ

Представлена информация о возможности применения оптических иллюзий для корректировки степени восприятия изображения человеком. Предлагается алгоритм использования оптических иллюзий с помощью современного программного обеспечения, а также приводится пример реализации подобной методики. Рассмотрены аспекты применения оптических иллюзий в рекламе и системах расширенной реальности.

Наличие в графических объектах специфических свойств, вызывающих искажение восприятия, позволяет использовать их для корректировки оптического восприятия изображений.

К подобным специфическим свойствам можно отнести оптические иллюзии, которые можно послойно структурировать на имеющемся изображении. Послойная топология содержит отдельные графические примитивы, привязанные к топологическим координатам объектов. Разложение каждого из типов иллюзий на примитивы определяется возможностями его влияния на восприятие. Кроме того большое значение имеет определение векторов направлений перемещения примитивов.

Каждый тип оптических иллюзий идентифицируется набором примитивов и стратегией изменения амплитуды воздействия [1]. Например, изменением толщины линий, интенсивностью цвета или взаимным расположением объектов, составляющих оптическую иллюзию.

У каждой группы иллюзий изменение степени восприятия реализуется различными методами. Кроме того, нельзя получить усиление восприятия простым масштабированием объектов. Зачастую некоторые типы иллюзий, присутствующие на одной картине, могут ослаблять или усиливать степень восприятия общей иллюзорности, это накладывает ограничения на выбор слоя и компоновку графических примитивов и заливок, формирующих иллюзию [3]. Для понимания алгоритма, с помощью которого можно применить оптические иллюзии для уже готового графического изображения, на рис. 1 представлена схема шагов корректировки изображения с использованием примитивов оптических иллюзий.



Рис. 1. Алгоритм корректировки изображения

Для создания оптических иллюзий можно использовать стандартные инструменты современных программных пакетов графического назначения таких как «Photoshop» и «CorelDraw». К стандартным инструментам можно отнести, к примеру, «масштабирование», «рисование фигур», «рисование линий», «изменение цвета» и т.д. Данные функциональные возможности просты в использовании и доступны как для профессионалов так и для обычных пользователей.

В качестве примера использования описанной выше технологии можно привести дизайн логотипов социальной сети LinkedIn. Данный логотип создан при помощи технологии перерегулирования [2]. Данная технология заключается в изменении размера отдель-

ных символов для достижения оптического баланса. Принцип перерегулирования был разработан в связи с тем, что, позиционирование каждого символа на основе его высоты обычно делает слово визуально непропорциональным и соответственно изменяет восприятие изображения [3]. Решение проблемы пропорциональности очень важно, к примеру, для сферы рекламы, так как в данной сфере очень важен визуальный эффект на потенциального покупателя. На представленном рис. 2 на логотипе «LinkedIn» символы «n», «e», «d» выходят за пределы базовой линии (А) и общей высоты. Обычно художникам-шрифтовикам приходится вручную оптически настраивать каждую пару символов для достижения наилучшего результата, так как на сегодняшний день не существуют программного обеспечения для автоматической корректировки изображения с помощью оптических иллюзий.



Рис. 2. Логотип социальной сети LinkedIn

Для автоматизации данного процесса было бы удобно использовать алгоритм, описанный ранее на рис. 1. В табл. 1 описано применение алгоритма корректировки изображения для исходного изображения логотипа «LinkedIn» (до применения принципа перерегулирования вручную).

Таблица 1

Ход работы алгоритма корректировки изображения при помощи оптической иллюзии

| Иллюзия | Геометрическая иллюзия |
|----------------------------------|--|
| Примитивы | Верхняя часть символа прямая Верхняя часть символа закругленная Нижняя часть символа прямая Нижняя часть символа закругленная |
| Топологические привязки | Прямая А, проходящая по верхней части символа Прямая В, проходящая по нижней части символа |
| Стратегия управления примитивами | Каждый символ с закругленной верхней частью увеличить на 2 мм по верней части Каждый символ с закругленной нижней частью увеличить на 2мм по верней части |

Влияние преобразованного с помощью оптических иллюзий изображения можно объяснить механикой работы человеческого глаза, а так же психофизиологическими особенностями восприятия графических изображений и текста. Практически любой графический объект, картина, видеоряд, обладают элементами иллюзорности с различной степенью воздействия на восприятие и психику человека.

На сегодняшний день не существует реальных программ для обработки изображений при помощи оптических иллюзий, именно поэтому данный вопрос является актуальным на сегодняшний день. Для идентификации топологических свойств отдельных оптических иллюзий, графических объектов, хорошо подходят искусственные нейронные сети, решающие задачу распознавания образов. Реализация данных подходов позволяет ввести новый инструмент редактирования, управляющий уровнем восприятия в системах дополненной реальности и рекламе.

Библиографический список

1. Двойственные миры Эшера. Оптическая иллюзия. – URL: <http://subscribe.ru/archive/> (дата обращения: 14.04.2018).
2. Официальный сайт «Ideaspectrum». – URL: <https://www.ideaspectrum.com/> (дата обращения: 02.09.2018).

3. Валько, А. А. Методы трансформации оптического восприятия объектов / А. А. Валько // Новые информационные технологии и системы : сб. науч. ст. VIII Междунар. науч.-практ. конф. – Пенза : Изд-во ПГУ, 2016. – С. 65–69.

М. А. Вахромеев, А. Д. Глебов

Пензенский государственный университет, Пенза, Россия

НЕЙРОННЫЕ СЕТИ С ИЗМЕНЯЕМОЙ ТОПОЛОГИЕЙ И ОСОБЕННОСТИ ИХ ОБУЧЕНИЯ

Представлено описание нейронных сетей с изменяемой топологией, их возможности и современное развитие: что это за вид нейронных сетей, как они обучаются, генерируются и где их можно использовать в наше время.

Введение

Нейронные сети – это определенные структуры так называемых нейронов, с помощью которых можно выполнять самые различные задачи в области человеческой деятельности. Они создавались с целью максимально приближенно воссоздать работу человеческого мозга по аналогии с его нейронами. Впервые о нейронных сетях заговорили в 1940-х годах, с тех пор их развитие претерпело как взлеты, так и падения. Сейчас же наблюдается большой всплеск к алгоритмам машинного обучения в самых различных областях человеческой деятельности, от самых простых систем, решающих тривиальные задачи, до очень сложных задач управления различными технологичными процессами.

Многие, кто слышал что-то о нейронных сетях, представляют их как определенную структуру нейронов со связями между ними, связи могут иметь различные веса, каждая сеть имеет входной и выходной слои нейронов, и топология сети фиксирована. Конечно, в этом нет ничего ошибочного, но такие нейронные сети не являются «гибкими», их сложно адаптировать под выполнение конкретной задачи, условия которой могут изменяться. Поэтому в последнее время стали появляться и развиваться нейронные сети с изменяемой топологией, о которых мы будем говорить далее.

Важные особенности нейронных сетей с изменяемой топологией

Нейронные сети с изменяемой топологией (TWEANN – Topology Weight Evolving Artificial Neural Networks) – это новый вид нейронных сетей, появившихся сравнительно недавно. Их реализация основана на теории эволюции и генетических алгоритмах. Как правило, процесс моделирования и обучения идет сразу на нескольких агентах, постепенно выбираются лучшие из них, самые результативные, значения фитнес-функции которых выше, выбранные агенты могут скрещиваться друг с другом, а в процессе скрещивания к ним, как правило, применяются различные мутации.

За счет того, что обучение такого вида нейронных сетей происходит с помощью эволюционных алгоритмов, они могут быстро адаптироваться и обучаться под выполнение конкретной задачи, условия которой непостоянны. Эволюционные алгоритмы позволяют производить над агентами среды моделирования самые различные операции, схожие по своему виду с биологическими процессами, естественный отбор, мутации, воспроизводство и т.д. Таким образом, моделируя алгоритм работы для выполнения какой-либо задачи, мы будем оперировать только самыми сильными агентами, которые показывают наилучший результат, другими словами, «выживают» сильнейшие. За счет такого подхода достигается очень высокая гибкость и результативность, а конкуренция между агентами приводит к ускорению процесса обучения.

Помимо всего сказанного, эволюционные алгоритмы избавляют исследователей от проблемы постоянной настройки искусственных нейронных сетей, если бы результат работы какого-то алгоритма не удовлетворял условиям задачи, приходилось бы корректировать

веса нейронов, изменять связи между ними, менять структуру сети и т.д. Все эти задачи могут выполнять эволюционные алгоритмы с успехом не меньшим, чем если бы это делал человек, к тому же гораздо быстрее.

В самом начале обучения нейронные сети могут задаваться случайно, либо быть вообще пустыми, если решаемая проблема довольно проста. Для заданных структур ставится определенного вида задача, которую они должны решить, и в процессе работы для каждого агента рассчитывается значение фитнес-функции, сети с наибольшим значением продолжают обучение, остальные агенты отбрасываются.

К тому же, зачастую, в самой начальной стадии процесса моделирования и обучения нейронной сети мы просто не знаем ничего о том, как настраивать веса, связи и т.д. А эволюционные алгоритмы показывают здесь хорошие результаты, поскольку уже на самых первых раундах обучения происходит отбор наиболее результативных агентов и можно судить об успешном начале процесса обучения.

Авторами предлагается использование эволюционных алгоритмов для моделирования и обучения TWEANN-сетей, как самых перспективных в настоящее время.

Преимущества использования эволюционных алгоритмов при обучении нейронных сетей с изменяемой топологией

Эволюционные алгоритмы предоставляют в себе все необходимое, чтобы быстро обучать и моделировать нейронные сети для выполнения конкретной задачи. Если на начальном этапе происходит отбор из множества популяций, то скорость обучения заметно возрастает, и результат будет достигнут заметно быстрее, хотя на это потребуются дополнительные вычислительные ресурсы.

Как и в биологических процессах, эволюция приводит к спонтанному появлению самых необычных признаков у живых организмов, которые оказываются полезны при их дальнейшем выживании, переводя это на язык информационных технологий, можем провести аналогию, когда обучаемые агенты – отдельные нейронные сети, после нескольких раундов обучения ведут себя по-разному. Каждый отдельный из них показывает те или иные особенности, а по результатам сравнения значений их фитнес функции мы можем отбирать сильнейших для продолжения обучения, проделывая над ними различные операции скрещивания, мутации и т.д. При этом процессе сохраняются появляющиеся у агентов признаки, которые поспособствовали их прогрессу в решении конкретной прикладной задачи.

Самой главной особенностью нейронных сетей, которые обучаются с применением эволюционных алгоритмов, является то, что они сами корректируют себя, сети получаются как компактными – для решения простых задач, либо большими – для реализации сложных решений. Частный случай нейроэволюции, благодаря такой реализации, решения получаются более компактными и оптимизированными, в дальнейшем они будут использовать заметно меньше вычислительных ресурсов, следовательно – скорость работы таких нейронных сетей значительно выше.

Вывод

Основной плюс эволюционных алгоритмов в том, что при происходящих мутациях на «кусках» решений могут возникать неожиданно эффективные результаты. Чего не случится при использовании «метода обратного распространения ошибки» или при обучении с подкреплением, так как они привязаны к локальному градиенту и оперируют меньшим пространством решений. Но с другой стороны, для этого требуется производить большее количество вычислений, нужны более серьезные вычислительные ресурсы. К счастью, в связи с последним развитием микропроцессоров, различных архитектур и прочего, данная проблема уже не является такой обременяющей и не сдерживает дальнейшие исследования в данной области.

Библиографический список

1. Галушкин, А. Нейронные сети. Основы теории / А. Галушкин. – М. : Горячая Линия-Телеком, 2012. – 496 с.
2. Гладков, Л. А. Генетические алгоритмы : учеб. пособие / Л. А. Гладков, В. В. Курейчик, В. М. Курейчик. – 2-е изд. – М. : Физматлит, 2006. – С. 320.

З. Афонин, А. Ю. Подходы к оптимизации обучения нейронных сетей с изменяемой топологией / А. Ю. Афонин, А. Д. Глебов, Л. М. Шехтман, К. А. Дудкин // Информационные технологии в науке и образовании. Проблемы и перспективы : сб. науч. ст. Всерос. межвуз. науч.-практ. конф. (г. Пенза, 14 марта 2018 г.) / под ред. Л. Р. Фионовой. – Пенза : Изд-во ПГУ, 2018. – С. 84.

Д. А. Голдуева¹, М. А. Мокшанина²

¹ Пензенский государственный университет, Пенза, Россия

² Пензенский государственный аграрный университет, Пенза, Россия

АНАЛИЗ ПОВЕРХНОСТИ ЗЕМЛИ С ПОЗИЦИИ СТОХАСТИЧЕСКОЙ ГЕОМЕТРИИ И ФУНКЦИОНАЛЬНОГО АНАЛИЗА

Рассмотрены вопросы применения метода, основанного на аппарате стохастической геометрии и функционального анализа, к проблеме распознавания объектов на поверхности земли по аэрофотоснимкам.

Современный уровень технического развития позволяет получать снимки поверхности земли с разной степенью детализации, что делает возможным решение широкого круга практических задач анализа объектов на поверхности земли. К подобным задачам можно отнести: анализ глобального изменения климата, управление территориальным лесным хозяйством, городское планирование застройки, задачи поиска объектов и многие другие.

В настоящей статье рассмотрена задача распознавания типов участков поверхности земли на снимках, полученных с помощью аэрофотосъемки. Пример анализируемых изображений приведен на рис. 1. К типам участков, подлежащим анализу, можно отнести: лес, поле, населенный пункт, река, озеро, дорога, и т.д.



Рис. 1. Изображение поверхности земли, полученное с помощью аэрофотосъемки

Для того чтобы распознать тип участка на изображении поверхности земли, необходимо предварительно провести сегментацию снимка с целью выявления однородных объектов. В настоящий момент существуют несколько достаточно эффективных методов сегментации 2D изображений, например, изложенный в работе [1].

После проведения процедуры сегментации изображение разбивается на области, каждая из которых представляет собой цветную текстуру. В литературе встречаются несколько разных определений понятия текстура. В настоящей работе под текстурой будет пониматься изображение поверхности с повторяющимися примитивами каким-либо образом распределенными на ней.

Существующие в настоящий момент методы анализа цветных текстур имеют два существенных недостатка. Во-первых, большинство из них предварительно упрощают анализируемые цветные текстуры, приводя их к полутоновому или даже к бинарному виду, теряя

при этом существенную часть информации, которую несет в себе цвет. Во-вторых, существующие в настоящий момент методы анализа цветных текстур оперируют небольшим количеством признаков, каждый из которых имеет конкретную интерпретацию в терминах рассматриваемой проблемы. В этом случае представляется весьма вероятным столкновение со значительными трудностями при формировании информативных признаков более сложных цветных текстур подобными методами. В настоящей статье использован подход к формированию признаков цветных текстур, основанный на аппарате стохастической геометрии и функционального анализа. Его преимущество, по сравнению с существующими на данный момент методами анализа цветных текстур, заключается в следующем. Во-первых, предлагаемый метод анализа цветных текстур на основе стохастической геометрии и функционального анализа позволяет в режиме автоматической генерации формировать десятки тысяч признаков исследуемых изображений (триплетных признаков) без участия эксперта-аналитика. Опора на большое количество признаков является решающей предпосылкой высокой надежности распознавания. Во-вторых, предлагаемый метод не предполагает предварительного упрощения анализируемых цветных текстур. Формируемая группа триплетных признаков характеризует как геометрические параметры, так и особенности цвета анализируемых цветных текстур, что позволяет более полно описать объекты исследования.

Согласно предлагаемому методу признаки изображений имеют структуру в виде композиции трех функционалов [2]

$$P(F) = \Theta \circ P \circ T f(\theta, \rho, t), \quad (1)$$

где ρ, θ – нормальные координаты сканирующей прямой $l(\rho, \theta)$, с которыми связаны функционалы P и Θ соответственно; функционал T связан с параметром t , задающим точку на сканирующей прямой $l(\rho, \theta)$; $f(\theta, \rho, t)$ – функция, характеризующая пересечение функции изображения со сканирующей прямой $l(\rho, \theta)$. В связи с характерной структурой такие признаки были названы триплетными.

При анализе цветных текстур использовалась модель представления цвета RGB, согласно которой цвет разделяется на три компонента: красный, зеленый и голубой. Таким образом, для распознавания цветных текстур, были выделены две группы триплетных признаков:

- 1) признаки, характеризующие геометрические особенности изображения;
- 2) признаки, характеризующие особенности цвета текстуры.

При этом вторая группа, в свою очередь, подразделяется на три подгруппы: признаки, характеризующие особенности красного, зеленого и голубого компонента цвета текстуры соответственно. Признаки первой и второй группы имеют одинаковую трехфункциональную структуру вида (1). Отличие между ними заключается лишь в подходе к заданию характеристик однородных по яркости отрезков сканирующих прямых. Для построения признаков, характеризующих геометрические особенности изображения, однородным по яркости отрезкам сканирующих прямых ставится в соответствие некоторая геометрическая величина (например, длина отрезка). Для построения признаков, характеризующих особенности цвета текстуры, однородным по яркости (того или иного компонента цвета) отрезкам сканирующих прямых ставится в соответствие некоторая яркостная величина.

Так как характеристики цвета изображения существенно зависят от степени освещенности объекта в момент фотосъемки, то для построения триплетных признаков второй группы, однородным по яркости отрезкам сканирующих прямых целесообразно поставить в соответствие некоторую относительную характеристику яркости отрезка. В качестве та-

кой характеристики может служить $I_i = \frac{\sum_{j=k}^m f(\theta, \rho, t_j)}{(m-k+1) \cdot \bar{I}_l}$, где t_k, t_{k+1}, \dots, t_m принадлежат одному однородному по яркости отрезку сканирующей прямой, t_{k-1}, t_{m+1} принадлежат другому отрезку той же прямой, или t_{k-1}, t_{m+1} не принадлежат сетчатке, \bar{I}_l – средняя яркость всех затемненных участков сканирующей прямой. В этом случае формируемые триплетные при-

знаки, характеризующие особенности цвета текстуры, не зависят от изменений освещенности исследуемых объектов в момент фотосъемки.

Для распознавания цветных текстур представляют особый интерес признаки, инвариантные по отношению к группе движений и линейным деформациям изображений, так как опора на инвариантные признаки придает гибкость, надежность и универсальность системам распознавания. При определенном выборе функционалов, входящих в структуру триплетного признака, можно получить характеристики цветных текстур, обладающие указанным свойством инвариантности. Например, признак $PI(F) = \Theta_I \circ P_I \circ T_I$ обладает свойством инвариантности к переносу, где

$$T_1 = \max_i x_i,$$

где $x_i = \sum_{j=k}^m f(\theta, \rho, t_j)$, $t_k = t_{k+1} = \dots = t_m = 1$, $t_{k-1} = t_{m+1} = 0$ или t_{k-1}, t_{m+1} не принадлежат сетчатке;

$$P_2 = \sqrt{\sum_{i=1}^m g^2(\theta_j, \rho_i)},$$

где $g(\theta_j, \rho_i) = T(F \cap I(\theta_j, \rho_i))$, m – число дискретных значений ρ ;

$$\Theta_7 = \sum_{j=1}^n \ln |h(\theta_j) + 1| \cdot \Delta\theta,$$

где $\Delta\theta$ – шаг сканирования по θ .

После описанной выше процедуры формирования признаков получаем большое количество абстрактных характеристик (десятки тысяч) анализируемых текстур, не все из которых эффективны. Поэтому далее была проведена минимизация признакового пространства, подробное описание которой приведено в [3]. После чего согласно выбранному решающему правилу [4], основанному на теории компактности, каждый из анализируемых участков изображения относился к тому или иному типу.

Проведенные эксперименты показали высокую эффективность применения описанного метода к проблеме распознавания типов участков поверхности земли на снимках, полученных с помощью аэрофотосъемки. Построенная группа признаков позволит достаточно полно описать цветные текстуры. Благодаря трехкомпонентной структуре триплетных признаков возможна генерация большого их количества, что позволяет увеличить гибкость, универсальность и надежность распознавания. Причем, как показывают проведенные эксперименты, при определенном выборе функционалов, входящие в структуру триплетного признака, формируемые характеристики приобретают свойства инвариантности к группе движений и линейным деформациям. Опора на большое количество признаков, описывающих как геометрические, так и яркостные особенности цветных текстур, позволяет повысить надежность их анализа.

Библиографический список

1. Чочиа, П. А. Сегментация изображений на основе анализа расстояний в пространстве признаков / П. А. Чочиа // Автометрия. – 2014. – Т. 50, № 6. – С. 97–110.
2. Федотов, Н. Г. Геометрические и яркостные триплетные признаки полутоновых изображений / Н. Г. Федотов, Д. А. Голдуева // Интеллектуализация обработки информации (ИОИ-2012) : тр. 9-й Междунар. конф. – Будва, Черногория, 2012. – С. 448–452.
3. Федотов, Н. Г. Минимизация размерности признакового пространства при распознавании полутоновых изображений со сложной текстурой / Н. Г. Федотов, Д. А. Голдуева // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2010. – № 1 (13). – С. 54–63.
4. Fedotov, N. G. Recognition of halftone textures from the standpoint of stochastic geometry and functional analysis / N. G. Fedotov, D. A. Mokshanina // Pattern Recognition and Image Analysis. – 2010. – Vol. 20, № 4. – P. 551–556.

Е. С. Серов, Л. В. Гурьянов

Пензенский государственный университет, Пенза, Россия

МЕТОД РЕШЕНИЯ ЗАДАЧ СПОРТИВНОГО ПРОГРАММИРОВАНИЯ НА ОСНОВЕ ПРЕФИКСНОГО ДЕРЕВА

Рассматривается применение префиксного дерева в рамках решения ряда задач спортивного программирования, а также приводится пример алгоритма поиска наименьшего «исключающего или» чисел с помощью префиксного дерева.

Префиксное дерево (англ. *trie*) – это структура данных, хранящая набор пар ключ-значение, где ключом может выступать любая строка [1]. Оно представляет собой корневое дерево, где каждое ребро помечено каким-либо символом таким образом, что строка, образованная конкатенацией символов в пути от корня дерева до конкретной вершины, является ключом. В таком случае в вершине, перед которой находится ребро с последним символом ключа, записывается присвоенное данному ключу (рис. 1).

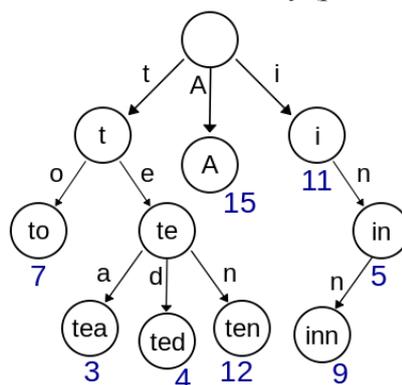


Рис. 1. Префиксное дерево, содержащее ключи «A», «to», «tea», «ted», «ten», «i», «in», «inn».

Поиск в такой структуре выполняется при помощи спуска от корня до искомой вершины, где каждый переход вниз к очередной вершине осуществляется по тому ребру, символ на котором равен очередному символу в ключе. Если после очередного спуска по ребру символы в ключе закончились, то алгоритм возвращает значение в текущей вершине. Если в какой-то момент оказывается, что ребра с символом равным очередному символу ключа не существует, это означает, что данный ключ не был добавлен в дерево.

Ребро префиксного дерева может быть помечено не одним символом, а сразу набором символов, что значительно ускоряет вычисления. Такое префиксное дерево называют сжатым префиксным деревом. Оно имеет огромную значимость в современном мире, так как используется в большинстве алгоритмов поиска.

Однако, несмотря на вычислительное превосходство сжатого префиксного дерева, существует класс задач, решаемый только обычным префиксным деревом. Одной из таких задач является поиск наименьшего XOR (исключающего или) в наборе чисел. Это одна из распространённых задач в спортивном программировании, а также достаточно частая задача в криптографии.

Формально задача ставится следующим образом: дано число C и множество чисел $M = \{M_1, M_2, M_3, \dots, M_n\}$, требуется найти такое число $X \in M$, чтобы результат операции $X \text{ XOR } C$ был минимален.

В то время, как использование простого перебора имеет сложность $O(n)$, а бинарное дерево поиска и подобные структуры не способны вернуть число с минимальным XOR, префиксное дерево позволяет решить данную задачу за $O(K)$, где K – количество бит в числе C . То есть при использовании 4-байтовых целочисленных типов данных K будет равно 32. Важно заметить, что для хранения числа C и всех чисел из M должны использоваться одинаковые типы данных.

Алгоритм поиска X работает следующим образом. Всё множество чисел M побитово запишем в префиксное дерево. При этом запись начнём со старшего бита, записав его в ребро сразу после корня, и будем спускаться по дереву вниз, записывая на каждое следующее ребро, очередной бит из числа. В итоге получится префиксное дерево глубиной K , где из каждой вершины есть максимум 2 ребра вниз (переход по символу '0' и по символу '1'). При этом в качестве строковых ключей используется бинарное представление чисел (иногда приходится добавлять лидирующие нули, чтобы все строки были длиной, ровно K), а в качестве значений в каждой вершине записано, сколько раз мы прошли через эту вершину в процессе добавления элементов. Это необходимо, чтобы в дальнейшем иметь возможность динамически удалять и добавлять новые числа в дерево.

Для поиска наименьшего XOR в построенной структуре данных надо перевести число C в бинарное представление, и при необходимости добавить лидирующие нули, чтобы количество битов в числе C стало равно K . После этого запускается поиск из корня дерева, где на каждой итерации будет необходимо выполнить следующие шаги:

1. Взять очередной бит числа C , обозначим его как B ;
2. Проверить, существует ли ребро с символом B ;
3. Если существует, то перейти по нему, иначе перейти по другому ребру;
4. Добавить к ответу символ, находящийся на ребре, по которому был осуществлён переход.

Эти действия повторить ровно K раз.

В итоге ответ будет выглядеть как строка длиной K , состоящая из символов '0' и '1'. Если перевести эту строку из бинарного представления в десятичное число, то мы получим элемент $X \in M$, и при этом $X \text{ XOR } C$ будем минимальным.

При реализации каждая вершина дерева описывается следующей структурой:

```
struct node{
    node* parent;
    vector<node*> child;
    int val, count;
    node(int val, node* parent){
        this->parent = parent;
        this->val = val;
        this->count = 0;

        child.resize(2, nullptr);
    }
};
```

В отличие от обычного префиксного дерева, где количество сыновей у вершины достаточно большое (мощность алфавита), у данного префиксного дерева в каждой вершине может быть максимум 2 ребра вниз, поэтому имеет смысла при создании вершины сразу выделить для них место (функция *resize*). Значение *nullptr* обозначает, что на данный момент ребра из этой вершины нет.

Поиск в таком дереве осуществляется при помощи следующего алгоритма:

```
int find(node* root, int mask){
    int ans = 0;
    for(int pos = K; pos >= 0; pos--){
```

```
int bit = get_bit(mask, pos);
if (root->child[bit] == nullptr){
    set_bit_1(ans, pos);
    bit = !bit;
}
root = root->child[bit];
}
return ans;
}
```

В данном примере K выступает константой и обозначает количество битов используемых для хранения чисел в памяти. Функция $get_bit(mask, pos)$ возвращает бит находящийся в числе $mask$ на позиции pos . Функция $set_bit_1(ans, pos)$ устанавливает в число ans на позицию pos бит, равный 1.

На вход функция $find$ принимает $node^* root$ – указатель на корень дерева и $intmask$ – число C , для которого необходимо найти минимальный XOR; возвращает – сформированное целое ans .

Приведенный алгоритм был успешно использован в решении задач, предлагаемых на этапах командного студенческого чемпионата мира по программированию ACM ICPC.

Библиографический список

1. Префиксное дерево // Википедия. – URL: https://ru.wikipedia.org/wiki/Префиксное_дерево (дата обращения: 23.09.2018).

Ю. И. Евсеева, А. С. Бождай, А. М. Бершадский

Пензенский государственный университет, Пенза, Россия

КОНЦЕПЦИЯ РЕФЛЕКСИИ И АРХИТЕКТУРА РЕФЛЕКСИВНОЙ ПРОГРАММНОЙ СИСТЕМЫ

Предложен новый подход к решению важной проблемы современной IT-индустрии – самоадаптации программных компонентов на основе концепции поведенческой рефлексии. Рассматриваются базовые принципы рефлексивной адаптации, а также идея использования инженерии линеек программных продуктов для создания адаптивных программ. Приводится обобщенное архитектурное решение для построения рефлексивных самоадаптивных систем, способных менять свои поведенческие характеристики непосредственно в ходе исполнения без перекомпиляции исходного кода. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00408.

Одним из наиболее интересных явлений в сфере психической деятельности разумного существа (человека) является рефлексия. Под рефлексией понимается способность мышления к критическому самоанализу. Благодаря ей, человек может не просто знать или чувствовать нечто, но и знать о своем знании и переживании. Это умение отслеживать то, что происходит на разных уровнях сознания, с возможностью дальнейшего переосмысления. Именно путем рефлексивного самоанализа человек способен выявлять те стратегии своего поведения, которые приводят к нежелательным результатам, а также формировать новые более эффективные [1, 2].

Одной из проблем современного программного обеспечения является проблема недостаточности знаний во многих предметных областях. Их динамика и сложность зачастую столь сложны, что невозможно учесть все связанные с ними обстоятельства на этапе разработки информационных моделей и компиляции исходного кода. Это заставляет создателей таких систем проявлять повышенное внимание к процессу сопровождения, регулярно выпускать обновления, не содержащие кардинально новой функциональности, но отнимающие внимание, время и материальные средства конечного пользователя [3].

Решением данной проблемы может стать внесение в функционал программной системы интеллектуальных свойств самоадаптации, схожих с механизмами человеческой рефлексии. В таком случае, разработчику достаточно формализовать лишь базовые модели анализа и поиска закономерностей (тем самым делая предположение, что какие-то моменты на этапе разработки остались неучтенными), а выявлением недостающих знаний система будет заниматься самостоятельно на протяжении своего жизненного цикла. Такой подход не только позволит снизить затраты на сопровождение системы, но и выявить те сведения о ее предметной области, которые могут быть неизвестны даже экспертам.

Концепция рефлексивной адаптации базируется на следующих принципах:

- Инвариантность к предметной области, типу и назначению программной системы;
- Масштабируемость;
- Интеллектуальный анализ ретроспективных данных.

Условимся называть данный тип адаптации рефлексивной самоадаптацией или рефлексией программной системы. Можно выделить три базовых уровня программной рефлексии (рис. 1): уровень самонаблюдения, самооптимизации и самоадаптации.

Уровень самонаблюдения является базовым и наиболее примитивно организованным уровнем. Он не описывается какой-либо сложной математической моделью. Основная задача механизмов адаптации, функционирующих на этом уровне - составление поведенческих протоколов программной системы.



Рис. 1. Уровни рефлексии программной системы

Уровень самооптимизации затрагивает организацию приложения на уровне его программного кода и интерфейса. Механизмы адаптации, работающие на этом уровне, используют поведенческую информацию уровня самонаблюдения для оптимизации указанных составляющих программной системы. В данном случае программа адаптируется не под конкретного пользователя или устройство, а устраняет в себе свойства, которые являются объективно нежелательными. Объективно нежелательными свойствами являются те, которые признаны таковыми для любых пользователей и при любых условиях. К числу таких свойств относится, например, избыточность пользовательского интерфейса.

Уровень самоадаптации является наиболее сложноорганизованным уровнем. Именно на этом уровне для формализации адаптивного поведения используются дополнительные модели - модели изменчивости. Модель изменчивости представляет собой своеобразное морфологическое множество адаптивной программной системы, т.е. множество его потенциальных конфигураций. Если уровень самооптимизации определяет то, насколько оптимально написан код или организован интерфейс, то уровень самоадаптации отвечает именно за поведенческую реорганизацию приложения в процессе выполнения. Именно на этом уровне формируются новые обратные связи системы.

В качестве моделей изменчивости предлагается использовать модели характеристик, в частности, их графическое представление – диаграммы характеристик (рис. 2).

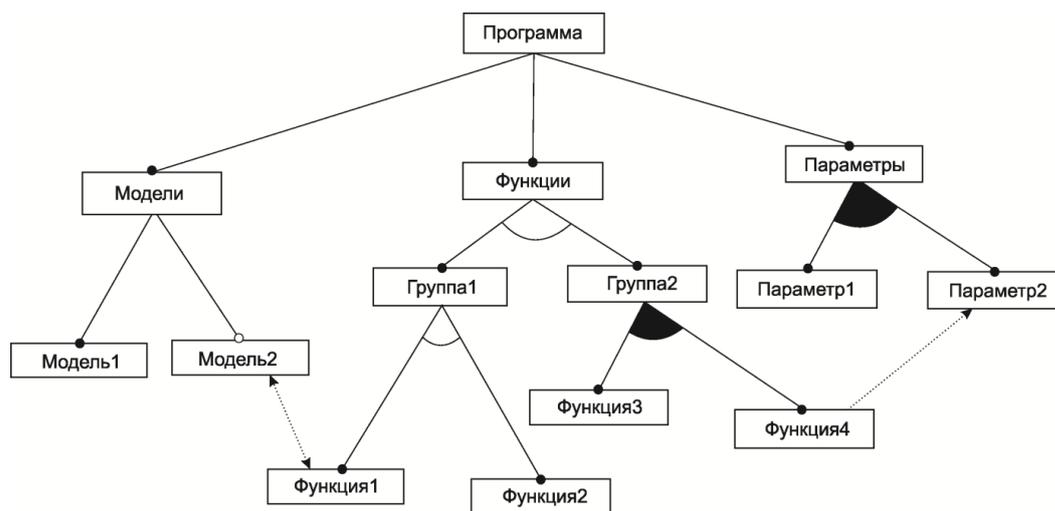


Рис. 2. Диаграмма характеристик

Диаграмма характеристик представляет собой усовершенствованное И/ИЛИ-дерево, вершинами которого являются компоненты программной системы, а связи между вершинами определяют совместимость компонентов в пределах одной конфигурации [4, 5, 6].

Рефлексивная программная система также должна иметь собственную специфическую архитектуру. Обобщая существующий опыт разработки систем с модульной архитектурой, можно сформулировать несколько важных принципов построения программных систем с рефлексивной самоадаптацией функционала:

- Минимальными компонентами программной системы, подверженными адаптации, являются блоки изменчивости. Каждый блок изменчивости, входящий в состав системы, отвечает за возможную реализацию какого-либо атомарного компонента программы (функции, класса, модели объекта предметной области и т.п.).
- Совокупность логически связанных блоков изменчивости, относящихся к одной предметной области и предназначенной для решения одной задачи, называется рефлексивным компонентом.
- Рефлексивная программная система может состоять из одного или нескольких рефлексивных компонентов. Их количество будет определять адаптивное разнообразие и рефлексивную сложность программы.
- Рефлексивный компонент включает в себя 2 базовых архитектурных уровня: уровень самооптимизации и уровень самоадаптации. Уровень самооптимизации имеет также четкое разделение на уровень представления (уровень интерфейса, не является обязательным) и уровень программного кода. В свою очередь, уровень самооптимизации построен на основе моделей изменчивости.
- Рефлексивные компоненты объединены в единую систему посредством общего интерфейса и изначально могут не иметь данных друг о друге. Однако в процессе рефлексивного анализа будет происходить выявление степени и характера влияния одних компо-

нентов на другие. Для этого необходимо предусмотреть каналы информационного обмена между компонентами.

Таким образом, общая математическая модель рефлексии адаптивной программной системы будет иметь два базовых уровня: уровень самооптимизации и уровень самоадаптации. Для построения математической модели уровня самоадаптации требуется использование моделей изменчивости. Оба уровня рефлексии также подразумевают использование алгоритмов интеллектуального анализа данных [7]: в случае с уровнем самооптимизации они будут использоваться для выявления объективно нежелательных свойств системы, в случае с уровнем самоадаптации – для формирования новых обратных связей.

Библиографический список

1. Гиппенрейтер, Ю. Б. Введение в общую психологию : курс лекций / Ю. Б. Гиппенрейтер. – М. : Юрайт, 2002. – С. 89.
2. Столяренко, Л. Д. Психология: учебник для вузов / Л. Д. Столяренко. – СПб. : Питер, 2016. – С. 124.
3. Control-based approach to balance services performance and security for adaptive service based systems (ASBS) / C. Liu, C. Jiang, H. Hu, K. Y. Cai, D. Huang, S. S. Yau // 33rd Annual IEEE International Computer Software and Applications Conference (COMPSAC'09). – Washington : IEEE Computer Society, 2009. – P. 473–478.
4. Berger, T. Variability modeling in the real: an empirical journey from software product lines to software ecosystems / T. Berger. – Leipzig: University of Leipzig, 2012. – 225 p.
5. Younis, O. Systems variability modeling: a textual model mixing class and feature concepts / O. Younis, S. Ghoul, M. H. Alomari // International Journal of Computer Science & Information Technology. – 2013. – № 5. – P. 127–139.
6. Webber, D. L. Modeling variability in software product lines with the variation point model / D. L. Webber, H. Goma // Science of Computer Programming. – 2004. – № 3. – P. 305–331.
7. Анализ данных и процессов / А. Барсегян, М. Куприянов, И. Холод, М. Тесс, С. Елизаров. – СПб. : БХВ-Петербург, 2009. – 314 с.

Ю. И. Евсеева, А. А. Гудков

Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ КОНЦЕПЦИИ ИНЖЕНЕРИИ ЛИНЕЕК ПРОГРАММНЫХ ПРОДУКТОВ ДЛЯ СОЗДАНИЯ АДАПТИВНЫХ ПРОГРАММНЫХ СИСТЕМ

Рассмотрены вопрос применения концепции инженерии линеек программных продуктов в разработке новой технологии синтеза адаптивных программных систем. Показана перспективность такого подхода. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 18-07-00408.

Инженерия линеек программных продуктов (Software Product Lines Engineering, SPLE) – это концепция повторного использования компонентов программного обеспечения, помогающая разрабатывать семейства (линейки) продуктов с сокращением времени выхода на рынок и повышением качества [1].

Центральным понятием концепции SPLE выступает понятие модели изменчивости (variability model). Модель изменчивости – это некоторое формализованное описание множества возможных конфигураций программной системы [2]. Наиболее близким по смыслу понятием в технических науках является понятие морфологического множества [3]. По сути, модель изменчивости и представляет собой морфологическое множество программной системы, дополненное некоторыми ограничениями и правилами. Данные ограничения и правила касаются вопросов совместимости отдельных компонентов потенциальной систе-

мы друг с другом, однако могут иметь и иной смысл (в зависимости от типа используемой модели изменчивости).

В SPLE используются следующие типы моделей изменчивости [4]:

- Модели характеристик – модели, которые наиболее часто используются на практике. Графически отображаются в форме модифицированного И/ИЛИ-дерева, называемого диаграммой характеристик (рис. 1), могут иметь различное формализованное представление (гиперграф, пропозициональная формула, алгебраическая нотация и др.).

- Ортогональные модели изменчивости. Схожи с моделями характеристик, графически также отображаются в форме диаграмм. Основное отличие заключается в том, что ортогональные модели показывают только наличие изменчивости в рассматриваемой программной системе, в то время как модели характеристик предоставляют более конкретное описание как предметной области, так и точек изменчивости.

- Модели решений. Модель решений включает в себя следующие компоненты [5]: вопросы из предметной области, на которые нужно получить ответы в процессе разработки программного продукта; множества возможных ответов на вопросы; ссылки на используемые компоненты (активы) или другие решения; описание последствий принятия решения (ответа на определенный вопрос или выбор определенного актива).

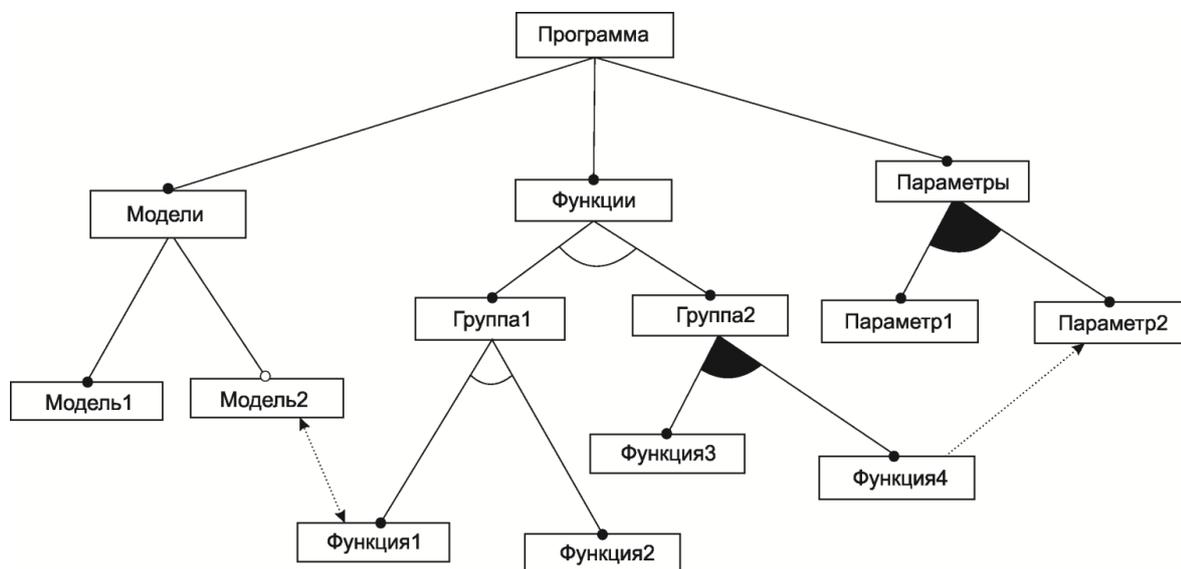


Рис. 1. Пример диаграммы характеристик

Как уже было сказано выше, в рамках концепции SPLE модели изменчивости представляют собой аналог моделей морфологических множеств. Процесс создания оптимальной конфигурации программного продукта по модели изменчивости схож с процессом структурно-параметрического синтеза технических систем по морфологическому множеству, но гораздо менее формализован.

Концепция DSPL [6] развивает идею SPLE: модели изменчивости динамических линеек программных продуктов (DSPL) используются для формирования оптимальной конфигурации программной системы в процессе ее выполнения. DSPL-системы можно считать адаптивными.

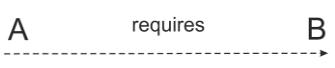
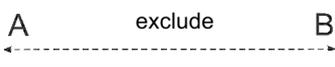
С нашей точки зрения, одним из перспективных подходов к реализации механизма программной рефлексии является интеграция принципов DSPL и технологии Data Mining [7]. При этом, первым шагом к созданию универсальной технологии синтеза адаптивных систем должна стать разработка математической модели изменчивости, задающей формальные правила структурных и функциональных изменений для широкого класса ПО. Для этих целей предлагается использовать хорошо известную модель характеристик (feature model) – она обладает хорошей визуальной наглядностью диаграммы (по этой причине, в

научном обиходе часто используются другие теримны – диаграмма характеристик, модель характеристик, feature diagram) и легко формализуема [8].

В табл. 1 представлены основные типы отношений в моделях характеристик. Каждая из характеристик служит абстракцией определенного программного компонента (модуля, класса, 3D-модели, мультимедийного ресурса и т.д.). Отношения между характеристиками определяют потенциальный набор конфигураций системы в целом. Кроме того, с помощью отношений будет задаваться степень совместимости различных компонентов в рамках одного программного продукта.

Таблица 1

Отношения в модели характеристик

| Обозначение | Наименование | Описание |
|---|--|--|
|  | Отношение обязательной дочерней характеристики | Если родительская характеристика включена в конфигурацию диаграммы, то дочерняя характеристика тоже должна быть включена |
|  | Отношение опциональной дочерней характеристики | Включение дочерней характеристики в конфигурацию не является обязательным |
|  | ИЛИ | Если родительская характеристика включена в конфигурацию, то также должно быть включено определенное количество дочерних |
|  | Исключающее ИЛИ | Только одна из представленных дочерних характеристик должна присутствовать в конечной конфигурации |
|  | Отношение включения | Включение в конфигурацию характеристики А требует включения характеристики В |
|  | Отношение исключения | Включение в конфигурацию характеристики А требует исключения характеристики В |

Для формального описания, зададим модель характеристик в виде ориентированного прямого гиперграфа (или F-графа):

$$F = (N, E, \Delta, \Psi), \quad (1)$$

где $N = \{N_1, N_2, \dots, N_n\}$ – конечное множество вершин графа, каждая из которых представляет собой характеристику диаграммы; $E = \{E_1, E_2, \dots, E_m\}$ – конечное множество гиперребер (отношений модели характеристик); $\Delta \in N$ – вершина, представляющая корневую характеристику диаграммы; $\Psi: E \rightarrow M, M \subset N \times N$ – функция маркировки, которая при-сваивает значение мощности $mv(E_i) = (min, max) \in M$, где M – конечное множество значений мощности модели характеристик, N – конечное множество вершин гиперграфа, каждому ориентированному гиперребру E_i , так что $min, max \in \mathbb{Z} \wedge min \geq 0, max > 0, min \leq max \wedge max \leq q_i$, где \mathbb{Z} – множество целых чисел.

Значение мощности – это пара из минимального и максимального количества вершин головного множества гиперребра, которые могут быть включены в конфигурацию диаграммы характеристик, которой соответствует гиперграф.

С помощью отношений опциональной дочерней характеристики, исключаящего ИЛИ и множественного ИЛИ реализуются блоки изменчивости – структурные единицы программной системы, затрагиваемые адаптивными процессами. Блок изменчивости может использоваться для определения множества состояний отдельного компонента программной системы, а также правила выбора того или иного состояния в каждом конкретном слу-

чае. Таким образом, можно описать, к примеру, ситуацию выбора оптимального алгоритма генерации изображения в зависимости от конфигурации аппаратной платформы или же настроить параметры какого-либо сервиса либо совокупности сервисов (системы).

Блок изменчивости может задавать стратегии адаптивное поведение отдельного компонента сервиса, всего сервиса, или даже всей системы в целом. Отсюда следует, что блок изменчивости не атомарен, он может включать в себя другие блоки, иметь иерархическую структуру.

Блок изменчивости описывается диаграммой характеристик (формула 2). Корневая вершина такой диаграммы называется точкой вариации.

$$VB = (SubF, BlockParams, BlockStates, Rule), \quad (2)$$

где $SubF$ – гиперграфовое (формула 1) представление модели характеристик блока изменчивости; $BlockParams = \{BlockParam_1, BlockParam_2, \dots, BlockParam_k\}$ – множество параметров, влияющих на состояние блока; $BlockStates = \{BlockState_1, BlockState_2, \dots, BlockState_z\}$ – множество состояний блока, при этом $BlockState_i \subseteq SubF \forall i = 1, \dots, z$; $Rule : BlockParams' \rightarrow BlockStates$ – функция, ставящая в соответствие каждому элементу множества $BlockParams'$ некоторый элемент множества $BlockStates$. Множество $BlockParams'$ представляет собой множество подмножеств $BlockParams$.

Множество влияющих параметров $BlockParams$ можно представить следующим образом:

$$BlockParams = ExtParams \cup InnerParams \cup TargetParams, \quad (3)$$

где $ExtParams = \{ExtParam_1, ExtParam_2, \dots, ExtParam_{k1}\}$ – множество внешних по отношению к рассматриваемому блоку параметров;

$InnerParams = \{InnerParam_1, InnerParam_2, \dots, InnerParam_{k2}\}$ – множество внутренних, варьируемых, параметров блока;

$TargetParams = \{TargetParam_1, TargetParam_2, \dots, TargetParam_{k3}\}$ – множество целевых параметров. Элементы множества $TargetParams$, как правило, связаны с некоторыми целевыми функциями и определяются (в отличие от других подмножеств множества $Params$) экспертным путем.

Библиографический список

1. The journey: a service-based adaptive serious game on probability / M. B. Carvalho, F. Bellotti, R. Berta, F. Curatelli, A. De Gloria, G. Gazzarata, J. Hu, M. Kickmeier-Rust, C. Martinengo // Serious games analytics / ed. by C. S. Loh, Y. Sheng, D. Ifenthaler. – Cham : Springer International Publishing, 2015. – P. 97–106.
2. Bernon, C. Engineering self-organizing systems / C. Bernon, V. Chevrier, V. Hilaire, P. Marrow // Self-organizing software: from natural to artificial adaptation / ed. by G. Di Marzo Serugendo, M. P. Gleizes, A. Karageorgos. – Heidelberg : Springer Berlin Heidelberg, 2011. – P. 283–312.
3. Waignier, G. A model-based framework to design and debug safe component-based autonomic systems / G. Waignier, A. F. Le Meur, L. Duchien // Architecture for adaptive software systems / ed. by R. Mirandola, I. Gorton, C. Hofmeister. – Heidelberg: Springer Berlin Heidelberg, 2009. – P. 1–17.
4. Torbeyns, J. Describing and studying domain-specific serious games / J. Torbeyns, E. Lehtinen, J. Elen. – Cham : Springer International Publishing, 2015. – 250 p.
5. Colledanchise, M. How behavior trees modularize robustness and safety in hybrid systems / M. Colledanchise, P. Ogren // 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. – Chicago : IEEE, 2014. – P. 53–62.
6. Game development tool essentials / P. Berinstein, R. Arnaud, A. Ardolino, S. Franco, A. Herubel, J. McCutchan, N. Nedelcu. – Heidelberg : Springer Berlin Heidelberg, 2014. – 467 p.
7. Kenett, R. S. Future directions of software cybernetics: A position paper / R. S. Kenett // 35th IEEE Annual Computer Software and Applications Conference Workshops. – Washington : IEEE Computer Society, 2003. – P. 43–44.

Е. А. Кольчугина

Пензенский государственный университет, Пенза, Россия

ТЕОРИЯ НУМЕРАЦИЙ КАК ОСНОВА ДЛЯ ПОСТРОЕНИЯ САМООРГАНИЗУЮЩИХСЯ ПРОГРАММНЫХ СИСТЕМ

Рассматриваются базовые основополагающие принципы теории нумераций и их связь с современными концепциями эволюционных и биоинспирированных вычислений. Устанавливается, что теория нумераций является краеугольным камнем для создания самоорганизующихся программных единиц и систем.

Основы теории нумераций были заложены в работах представителей научной школы А. Н. Колмогорова, в частности, В. А. Успенского [1], и развиты впоследствии в трудах А. И. Мальцева [2], Ю. Л. Ершова [3], В. М. Линькова [4].

Если изначально исследование нумераций было связано с понятием вычислимости, в частности, вычислимости программ для ЭВМ, то впоследствии открылся более широкий класс приложений, например, в области организации поиска и хранения информации в базах данных [4].

Ключевыми для теории нумераций являются понятия нумерации, нумерованного и номерного множеств, введенные в работах [1–4]. Все эти составляющие связываются одним соотношением [5]:

$$v_{Obj}: N_1 \rightarrow Obj,$$

где v_{Obj} – отображение нумерации, N_1 – номерное множество, как правило, подмножество натуральных либо целых неотрицательных чисел, Obj – нумерованное множество, то есть множество объектов, которым присвоены номера. Удобно рассматривать отображение v как функциональное и взаимнооднозначное. В этом случае можно ввести обратную к v_{Obj} функцию индексирования [5]:

$$ind_{Obj}: Obj \rightarrow N_1.$$

Одной из классических задач теории нумерации является задача нумерации числовой последовательности. В [3,4] такая нумерация рассматривается как результат вычисления n -аргументной функции свертки [3]:

$$s^1(x) = x,$$

$$s^2(x, y) = s(x, y),$$

...

$$s^n(x_0, \dots, x_n) = s(s^{n-1}(x_0, \dots, x_{n-1}), x_n).$$

Обратная задача решается при помощи n -развертки путем применения n одноместных функций [3]:

$$s^n(r_{n,1}(x), \dots, r_{n,n}(x)) = x, \quad r_{n,i}(s^n(x_0, \dots, x_n)) = x_i.$$

В [5, 6] автором этой статьи были определены операции восстановления, позволяющие, в отличие от n -развертки, получить всю исходную последовательность, а не один ее элемент.

Очевидно, что функции свертки, развертки и восстановления являются рекурсивными. С точки зрения организации корректного вычисления таких функций на ЭВМ, сворачиваемые последовательности необходимо расширить до $n+1$ элемента, добавив к ним в качестве последнего элемента длину сворачиваемой последовательности. Благодаря этому впоследствии можно обеспечить корректное восстановление исходной последовательности из свертки, поскольку будет известно необходимое количество применений рекурсивного правила.

Различные варианты функций свертки можно найти в [2,4]. Автором в [5,6] были предложены модифицированные варианты функции свертки и восстановления, ориентиру-

ванные на более эффективную реализацию на ЭВМ с точки зрения снижения роста значения функции свертки.

В дальнейшем следует ожидать, что интерес к теории нумерации будет возрастать в связи с развитием эволюционных и биоинспирированных вычислений.

Эти модели вычислений во многом исходят из аналогий с кодированным представлением генетической информации и ее наследственной передачей, наблюдаемыми в живой природе. Так, при помощи молекул нуклеиновых кислот, являющихся носителями наследственной информации в живых организмах, осуществляется кодирование всех синтезируемых организмом белков путем сопоставления каждой кодируемой аминокислоте, входящей в состав белка, набора нуклеотидов. Наиболее известен код молекулы ДНК, в котором используется алфавит из 4 кодирующих символов-нуклеотидов (A, T, G, C), триплетами которых, в свою очередь, кодируются аминокислоты. С учетом отношения комплиментарности между парами нуклеотидов $A-T$ и $C-G$, наблюдаемого в реальных биологических ДНК, в [7] было предложено рассматривать алфавит молекулы ДНК как вариант двоичного кодирования, используя символы $\{0, 1, \bar{0}, \bar{1}\}$.

В генетических алгоритмах и эволюционных вычислениях также часто используют кодирование представлений возможных решений. Каждое закодированное решение рассматривается как особь, представленная хромосомой, которая может репродуцироваться при участии других хромосом, возможно, с мутациями, т.е. случайными изменениями. Для кодирования генов в составе хромосом чаще всего используются целочисленные значения в двоичной записи. Таким образом, кодирование значений генов можно также рассматривать как нумерацию.

Автором в [8] был предложен способ кодирования генома самоорганизующейся программной единицы, сходный с используемым в генетических алгоритмах и эволюционных вычислениях. Самоорганизация – это один из ключевых принципов устройства реального мира [9]. Самоорганизующиеся системы, в частности, автокатакинетиические [10], можно охарактеризовать как совокупности динамически устойчивых отношений между потоками вещества-энергии. При этом компоненты, образующие систему, изначально поступают в нее извне и в конце концов удаляются из нее. Внутри системы компоненты могут подвергнуться не только замене, но и трансформации, распаду, перекомпоновке, в результате чего компоненты покинут систему в измененном виде. Именно благодаря необратимой трансформации компонентов удается сохранить устойчивость системообразующих отношений: в результате трансформации высвобождается свободная энергия, которая расходуется на совершение полезной работы против термодинамического равновесия, то есть на поддержание отношений между потоками.

Аналогичным образом самоорганизующаяся программная единица представляет собой совокупность отношений между потоками:

- потоком потребляемого процессорного времени как аналогом потока энергии;
- потоками обрабатываемых данных;
- потоками программных компонентов, в совокупности реализующих алгоритм, исполняемый самоорганизующейся программной единицей.

Все элементы, привносимые потоками в самоорганизующуюся программную единицу, подвергаются изменению или замене. Неизменными остаются только взаимоотношения между потоками, которые задаются с помощью структуры, называемой цифровой ДНК [5, 6, 8]. Исходя из того, что данные и фрагменты программного кода могут обобщенно рассматриваться как функции, а на множестве всех возможных функций можно задать нумерацию и каждой функции поставить в соответствие уникальный номер (индекс), цифровую ДНК можно представить в виде числовой последовательности. Номер, или индекс элемента последовательности, задает номер функции и определяет ее семантику. Сам элемент последовательности задает номер (индекс) текущей актуальной реализации функции. При изменении значений элементов последовательности изменяются реализации функций, то есть данные и программный код, образующие самоорганизующуюся программную единицу, но семантический аспект при этом не страдает, программа сохраняет смысл.

Множество таких самоорганизующихся программных единиц с переменными структурой и функциональностью образуют самоорганизующиеся программные системы с самоорганизацией по меньшей мере когерентного или даже континуального типа [9].

Как было показано в [6], с точки зрения разработки программ достаточно разработать одну последовательность, задающую цифровую ДНК, на основе которой в результате действий-аналогов размножения, совершаемых отдельными программными единицами, будет автоматически сгенерирован сложный самоорганизующийся программный комплекс.

К цифровым ДНК также можно применять преобразования сверток. В этом случае, как показано в [6], задача разработки программной системы может быть сведена к задаче получения некоего исходного числа, которое затем преобразуется путем восстановления в исходную цифровую ДНК.

Таким образом, самоорганизующаяся программная система, предназначенная, в сущности, для обработки чисел, сама является результатом выполнения последовательности операций над числами и числовыми последовательностями.

Библиографический список

1. Успенский, В. А. Вычислимые операции и понятие программы / В. А. Успенский // Успехи математических наук. – 1956. – № 11 (4). – С. 172–176.
2. Мальцев, А. И. Алгоритмы и рекурсивные функции / А. И. Мальцев. – 2-е изд. – М. : Наука, Гл. ред. физ.-мат. лит., 1986. – 368 с.
3. Ершов, Ю. Л. Теория нумераций / Ю. Л. Ершов. – М. : Наука, Гл. ред. физ.-мат. лит., 1977. – 416 с.
4. Линьков, В. М. Нумерационные методы в проектировании систем управления данными : монография / В. М. Линьков. – Пенза : Изд-во Пенз. гос. техн. ун-та, 1994. – 156 с.
5. Кольчугина, Е. А. Применение методов теории нумераций для представления эволюции кода программных агентов в темпоральных базах данных / Е. А. Кольчугина // Программная инженерия. – 2012. – № 4. – С. 19–22.
6. Кольчугина, Е. А. Теоретические основы построения самоорганизующихся программных систем с самоорганизацией континуального типа: дис. ... д-ра техн. наук / Кольчугина Е. А. – Пенза, 2017. – 295 с.
7. Паун, К. ДНК-компьютер. Новая парадигма вычислений : пер. с англ. / К. Паун, Г. Розенберг, А. Саломаа. – М. : Мир, 2004. – 528 с.
8. Кольчугина, Е. А. Структура цифрового организма в самоорганизующихся программных системах / Е. А. Кольчугина // Программные продукты и системы. – 2012. – № 2. – С. 51–54.
9. Руденко, А. П. Самоорганизация и синергетика / А. П. Руденко. – URL: <http://spkurdyumov.ru/what/samoorganizaciya-i-sinergetika-a-p-rudenko/> (дата обращения: 17.06.2018).
10. Swenson, R. Autocatakinetics, Yes – Autopoiesis, No: Steps Toward a Unified Theory of Evolutionary Ordering / R. Swenson // International Journal of General Systems. – 1992. – Vol. 21. – P. 207–228.

А. Ю. Тимонин

Пензенский государственный университет, Пенза, Россия

ЛЕКСИЧЕСКИЙ АНАЛИЗ ИСХОДНОГО НЕСТРУКТУРИРОВАННОГО ТЕКСТА В ПРОЦЕССЕ СОЦИАЛЬНОГО ПРОФИЛИРОВАНИЯ

Предлагается алгоритм решения задачи по извлечению семантического содержания из неструктурированного множества текстовых данных. В первую очередь новый метод используется для выделения социальных сущностей и определения эмоциональной окраски текста в процессе создания социальных профилей людей, но может найти свое применение и в других областях, таких как: анализ социальной напряженности, проверка авторства и т.д.

«Социальный профиль (СП) – это множество, состоящее из информации, способной тем или иным образом охарактеризовать человека, причем эта информация должна быть сгруппирована для удобства человеческого восприятия» [4]. Задача социального профилирования относится к сфере анализа социальных сетей, на основе разработанной математической модели СП строится результирующий социальный граф [1], отображающий социальные объекты, темы и явления (например, сообщества, местоположения, медиа-контент), а также отношения (со списком атрибутов) между ними. Исходные данные для создания социального профиля берутся из публичных источников сети Интернет. Они могут быть как текстовыми, так и нетекстовыми. В случае обработки неструктурированных текстовых данных возникает потребность получения структурированных выводов. Также анализ социальных данных с целью профилирования личностей имеет ряд особенностей [3]: на конечный результат влияет полнота информации об индивидуальных характеристиках человека (его деятельности, увлечениях, контактах, характере, репутации), исходные данные могут противоречить друг другу, иметь размытые границы детерминированности ввиду наличия синонимов, сокращений, слов с некорректным смыслом и ошибками в написании.

Общий алгоритм анализа исходных текстов СП включает 4 шага, представленные на рис. 1.

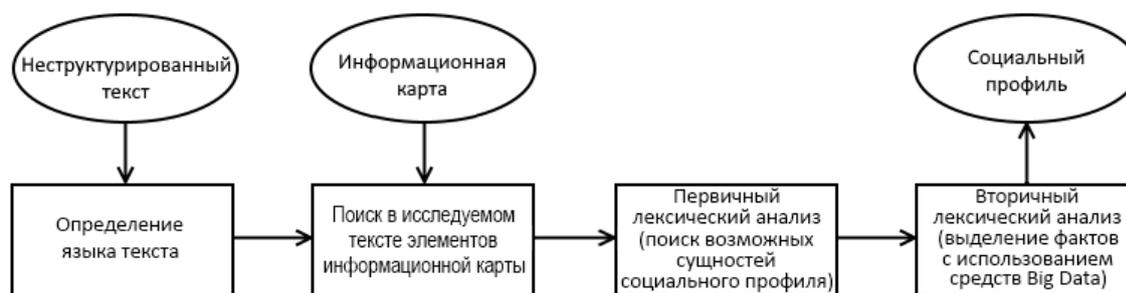


Рис. 1. Последовательность шагов анализа неструктурированного текста [2]

Данный алгоритм рассматривался в работе [2]. Первые два шага являются тривиальными, поэтому перейдем к рассмотрению этапов лексического анализа.

Первичный лексический анализ состоит в простой разбивке исходного текста на абстрактные логические элементы без выделения смысловой части. Его целью является нахождение кандидатов на роль социальных объектов и связей профиля. Для достижения этой цели проводится синтаксический разбор текста. Представим его обобщенный алгоритм:

– Разделение текста на логические части. Текущий шаг подразумевает параллельную обработку исходных данных для сокращения времени получения конечных результатов. Текст разделяется по таким критериям, как источник, время получения данных, язык (преобладания в части текста символов латиницы, кириллицы и т.д.). В случае необходимости разделения куска текста, к которому не применимы вышеназванные критерии, производится разбивка по приоритету символов:

1. Знаки конца предложения (точка, многоточие, вопросительный и восклицательный знаки), символ новой строки.
2. Знаки, разграничивающие части предложений (запятая, точка с запятой, двоеточие, тире), кавычки, скобки, знаки табуляции.
3. Пробелы.
4. Все прочие символы.

– Поиск грамматических основ. Происходит разбиение кусков текста на отдельные словоформы и последующее сравнение частей этих словоформ с эталонными значениями (например, для русскоязычных слов проверка окончаний) для определения частей речи и их характеристик: времени, лица, падежей и т.д. Найденные глаголы помечаются как кандидаты в сущности СП типа «связь» и соединяются с существительными (местоимениями,

числительными) в именительном падеже, выступающими кандидатами в категорию «объект» СП.

– Выявление эмоциональной окраски, восклицаний. Проводится поиск знаков восклицания, псевдографики (текстовые смайлы, эмоджи), сравнение словоформ со значениями из толкового словаря и присвоение им коэффициента настроения (-1 – негативный, 0 – нейтральный, 1 – положительный).

– Определение наклонов сказуемых. Для заполнения базы данных социальных объектов и связей в первую очередь важно рассматривать повествовательные части текста, т.к. только они могут представлять факты, в отличие от вопросительных и побудительных. Подлежащие являются кандидатами в таблицу «объектов», сказуемые – в таблицу Y. Остальной текст используется для выделения свойств объектов и настроения персоны/группы.

– Выделение в исходном тексте определений (как свойств кандидатов в таблицу «объектов»), обстоятельств (как свойств или реже кандидатов в таблицу Y), дополнений (как кандидат в таблицу «объектов»). Это достигается за счет установления принадлежности оставшихся словоформ частям речи через анализ морфем.

Результатом данного этапа является лингвистическая сеть, которую можно использовать в качестве основы для социального графа СП.

Вторичный лексический анализ заключается в группировке элементов полученной на предыдущем этапе лингвистической сети. Данный этап может представлять сложность для автоматической обработки. Существует проблема с определением семантики для омонимов, словоформ, определяющих одновременно разные объекты (антропонимы, топонимы и другие имена собственные, аббревиатуры, сокращения, имена нарицательные, которые обозначают отдельный объект внутри одноименной категории и т.д.) и наоборот, определяющих один объект, для которых требуется база данных синонимов. Возможным решением является использование средств deep learning, обученных обработке масштабных объемов текстовых данных. Окончательное заполнение таблиц происходит следующим образом:

– Осуществляется поиск предложений со словесными конструкциями из информационной карты или заданными вручную ключевыми словами, которые выступают в роли подлежащих или реже дополнений, если они связаны непосредственно с подлежащим – другим социальным объектом. Сказуемые этих предложений становятся связями СП, а дополнения новыми социальными объектами. Пример практической реализации данного этапа показан на рис. 2.

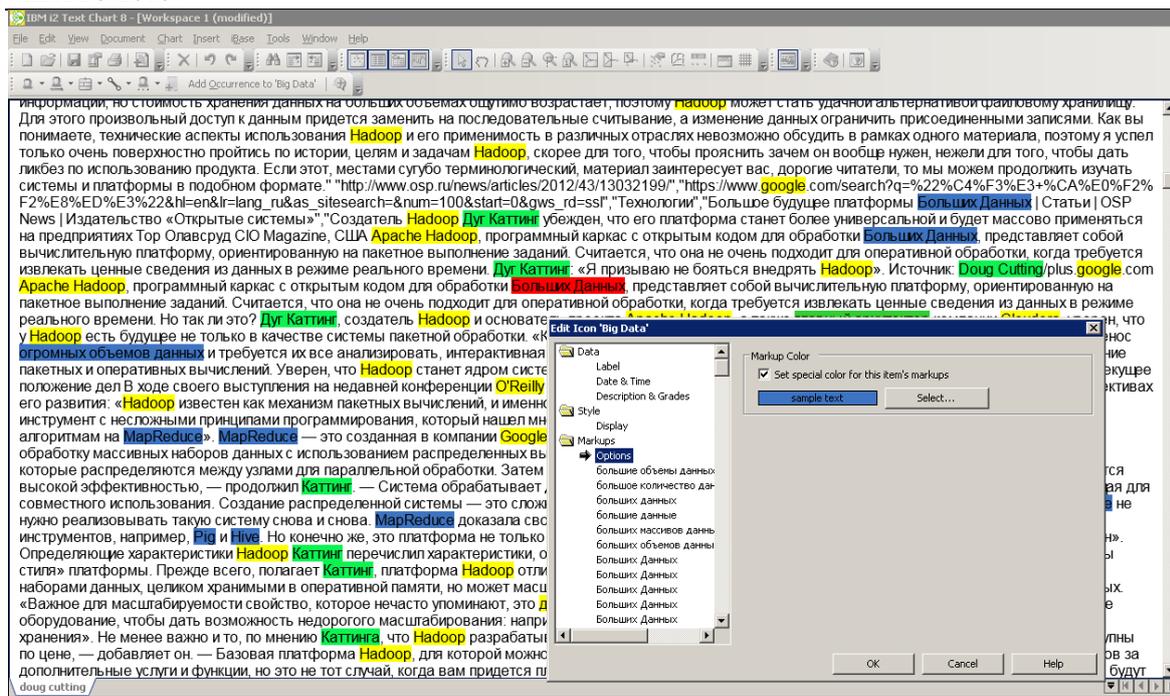


Рис. 2. Реализация выделения ключевых социальных сущностей в IBM i2 Text Chart, использующий средства визуального анализа

– Далее проводится повторный обход всего текста, учитывающий найденные на предыдущем шаге объекты и связи. На глубину обхода внутри исходных данных необязательно устанавливать ограничение, так как используются специализированные средства Big Data [5], которые устойчивы к высоким нагрузкам.

– Определяется конечное настроение социальных сущностей, как совокупность коэффициентов всех атрибутов и синонимичных значений каждого объекта. При создании СП в отдельную сущность нужно поместить настроение (sentiment) персоны или группы. Оно сильно зависит от времени и в отличие от прочих социальных сущностей, таких как хобби, работа, контактная информация, может меняться несколько раз в очень короткий промежуток времени (день, неделя). В то же время настроение является очень полезной величиной в анализе социума с использованием Больших данных, у которых одной из ключевых характеристик является скорость (velocity) появления и обработки данных.

Итоговая характеристика настроения социального профиля вычисляется как совокупный результат следующих действий:

– Анализ тональности текстов авторства рассматриваемой персоны. Поиск эмоционально окрашенных слов и конструкций (положительная/отрицательная/вопросительная/нейтральная окраска) с использованием словаря и выделением в тексте цитат, междометий, задающих интонации знаков препинаний, и наклонений глаголов.

– Анализ содержимого мультимедиа (запись голоса, изображений исследуемого человека), биометрии. Оценка тональности дается либо экспертом в ходе ручной проверки, либо средствами машинного обучения.

– Анализ связанного контента (например, определение яркости и цветовой гаммы изображений, жанровой принадлежности прослушиваемой музыки, пользовательских оценок). Решается за счет использования средств машинного обучения и нечеткой логики.

– Сравнение и корректировка настроения персоны с учетом общего фона окружающей среды (событий и связанных персон). Глубина обхода социального графа должна составлять два шага.

– Отдельно определяется настроение (отношение) к связанным объектам социального профиля: поиск эмоционально окрашенных словоформ в описаниях свойств у объектов и связей СП.

Настроение группы является усредненным значением настроений всех членов группы. Тональность по отношению к некоторой социальной группе определяется через поиск эмоционально окрашенных словоформ в описании всех направленных к данной группе связей как их сумма. Отношение некоторой социальной группы к рассматриваемой персоне или другой группе определяется аналогично – как сумма связей, направленных от данной группы.

Результатом данного этапа после слияния в социальные сущности и установки границ применимости для прикладных задач является полноценный СП. В рамках прикладных задач аналитик выбирает нужный ему набор вершин графа и определяет для них порядки окрестностей.

Заключение

В рамках данной работы были рассмотрены этапы первичного и вторичного лексического анализа неструктурированного текста в процессе построения социального профиля человека. Были приведены алгоритмы синтаксического анализа неструктурированного текста, определения тональности и выявления семантических зависимостей для категоризации исходных данных. В дальнейших исследованиях планируется связать представленные подходы с методами аналитики мультимедийных данных для обеспечения более глубокого анализа в задаче социального профилирования. Предполагается реализация показанных алгоритмов для человеко-машинной обработки на основе решений Big Data.

Библиографический список

1. Анализ социальных сетей // Материал из Википедии – свободной энциклопедии. – URL: https://ru.wikipedia.org/wiki/Анализ_социальных_сетей (дата обращения: 24.08.2017).
2. Бождай, А. С. Методы анализа гетерогенных данных для построения социального профиля / А. С. Бождай, А. Ю. Тимонин // Russian Journal of Management. – 2017. – Vol. 5, № 3. – С. 481–489.
3. Бождай, А. С. Требования к типу и содержанию данных социального профиля человека / А. С. Бождай, А. Ю. Тимонин // Актуальные вопросы современной науки: теория и практика научных исследований : сб. науч. тр. Всерос. науч.-практ. конф. – Пенза : Изд-во Пенз. гос. технол. ун-та, 2017. – С. 257–260.
4. Тимонин, А. Ю. Разработка социального профиля на основе имеющихся инструментальных средств / А. Ю. Тимонин // Новые информационные технологии и системы : сб. науч. ст. XII Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2015. – С. 221–224.
5. Шмид, А. В. Новые методы работы с большими данными: победные стратегии управления в бизнес-аналитике / А. В. Шмид // Научно-практический сборник / под ред. д.т.н., проф. А. В. Шмида. – М. : ПАЛЬМИР, 2016. – 528 с.

А. Е. Хохлов

Пензенский государственный университет, Пенза, Россия

МОДЕЛИРОВАНИЕ ВЫНУЖДЕННЫХ КОЛЕБАНИЙ ПЛАСТИН НА КЛЕТОЧНЫХ НЕЙРОННЫХ СЕТЯХ

Рассматривается дифференциальное уравнение в частных производных четвертого порядка, описывающее вибрацию пластин под воздействием внешней силы. Конечно-разностная аппроксимация по пространству и времени этого уравнения приводит к разностным схемам в нестационарной постановке. Доказывается возможность решения этих схем на клеточных нейронных сетях.

Одним из основных элементов различных конструкций является плоская пластина, свободно опертая или жестко закрепленная по краям и подверженная различного рода ди-

намическим нагрузкам. При решении таких задач традиционно используют идеи параллелизма, так как применение обычных ЭВМ требует значительных временных затрат.

Рассмотрим дифференциальное уравнение бигармонического типа, описывающее вибрационные процессы в пластинах [1]

$$\nabla^4 \varphi = \frac{\partial^4 \varphi}{\partial x^4} + 2 \frac{\partial^4 \varphi}{\partial x^2 \partial y^2} + \frac{\partial^4 \varphi}{\partial y^4} = f(t) - d \frac{\partial^2 \varphi}{\partial t^2}, \quad (x, y) \in S, \quad (1)$$

с граничными условиями жесткого закрепления

$$\varphi = 0, \quad \frac{\partial \varphi}{\partial n} = 0, \quad (x, y) \in \Gamma, \quad (2)$$

и свободного (шарнирного) опирания

$$\varphi = 0, \quad \frac{\partial^2 \varphi}{\partial n^2} = 0, \quad (x, y) \in \Gamma, \quad (3)$$

где $\varphi(x, y)$ – отклонение точек пластины (искомое решение), $f(t)$ – внешняя сила, d – коэффициент, характеризующий материал пластины, S – прямоугольная область решения, Γ – граница области.

В нестационарной постановке для решения такого уравнения кроме граничных условий должны быть заданы и начальные условия при $t = 0$

$$\varphi(0) = p \quad \frac{\partial \varphi}{\partial t} = q.$$

Конечно-разностная аппроксимация по пространственным переменным сводит задачу (1) к системе обыкновенных дифференциальных уравнений по времени

$$A\varphi = f(t) - \gamma \frac{\partial^2 \varphi}{\partial t^2}, \quad (4)$$

где A – конечно-разностный оператор бигармонического уравнения, $\gamma = dh^4$, h – шаг аппроксимации по пространству.

Каждый элемент вектора $A\varphi$ имеет вид:

$$20\varphi_{k,l} - 8(\varphi_{k-1,l} + \varphi_{k,l-1} + \varphi_{k+1,l} + \varphi_{k,l+1}) + 2(\varphi_{k-1,l-1} + \varphi_{k+1,l-1} + \varphi_{k-1,l+1} + \varphi_{k+1,l+1}) + (\varphi_{k+2,l} + \varphi_{k,l-2} + \varphi_{k,l+2} + \varphi_{k-2,l})$$

Для конечно-разностной аппроксимации задачи по времени введем в рассмотрение систему узловых точек $t_{j+1} = t_j + \tau$, где τ – шаг аппроксимации по времени, j – номер шага по времени. Можно предложить три разностные схемы в матричной форме, аналогичные приведенным в [2]:

1) явная схема

$$\gamma \frac{\varphi^{j+1} - 2\varphi^j + \varphi^{j-1}}{\tau^2} + A\varphi^j = f^j \quad (5)$$

2) неявная схема

$$\gamma \frac{\varphi^{j+1} - 2\varphi^j + \varphi^{j-1}}{\tau^2} + A\varphi^{j+1} = f^j \quad (6)$$

3) неявная схема (аналог схемы Кранка-Николсона)

$$\gamma \frac{\varphi^{j+1} - 2\varphi^j + \varphi^{j-1}}{\tau^2} + \frac{1}{2}(A\varphi^{j-1} + A\varphi^{j+1}) = f^j \quad (7)$$

Для всех трех схем должны быть заданы начальные условия

$$\varphi^0 = p \quad \varphi^1 = p + q\tau - \frac{\tau^2}{2\gamma}(Ap - f^0) \quad (8)$$

Для явной схемы (5) решение на каждом последующем временном шаге $j+1$ находится из решения на двух предыдущих шагах j и $j-1$

$$\varphi^{j+1} = (2E - \frac{\tau^2}{\gamma} A)\varphi^j - \varphi^{j-1} + \frac{\tau^2}{\gamma} f^j, \quad (9)$$

где E – единичная матрица.

Эта рекуррентная формула легко реализуется на ЦВМ. Однако при её использовании могут накапливаться ошибки. Чтобы ослабить влияние ошибок, можно рассчитывать на каждом шаге не очередное приближение к решению φ^{j+1} , а приращение решения $\Delta\varphi^{j+1}$. Само решение можно затем рассчитать по формуле

$$\varphi^{j+1} = \varphi^j + \Delta\varphi^{j+1} \quad (10)$$

В этом случае формулу (9) можно представить так

$$\Delta\varphi^{j+1} = (2E - \frac{\tau^2}{\gamma} A)\Delta\varphi^j - \Delta\varphi^{j-1} + \frac{\tau^2}{\gamma} \Delta f^j, \quad (11)$$

где $\Delta\varphi^j = \varphi^j - \varphi^{j-1}$, $\Delta\varphi^{j-1} = \varphi^{j-1} - \varphi^{j-2}$, $\Delta f^j = f^j - f^{j-1}$,

Следует заметить, что в расчетах по такому алгоритму, величины $\Delta\varphi$, f , φ , Δf – векторные и их расчет происходит с помощью последовательного обхода узловых точек на каждом временном интервале. Это является недостатком явной схемы. Другим недостатком следует признать условную устойчивость явной схемы, т.е. существование некоторого ограничения на длительность интервала между соседними шагами.

От этих недостатков свободны неявные схемы. Из неявной схемы (6) можно получить выражение для расчета φ^{j+1}

$$\left(A + \frac{\gamma}{\tau^2} E \right) \varphi^{j+1} = 2\frac{\gamma}{\tau^2} \varphi^j - \frac{\gamma}{\tau^2} \varphi^{j-1} + f^j \quad (12)$$

Для расчета необходимо обращение матрицы, что на ЦВМ представляет значительные трудности. Схема (12) легко разрешима на клеточных нейронных сетях (CNN) [3] и абсолютно устойчива. Причем матрица $A + \frac{\gamma}{\tau^2} E$ всегда положительно определена, так как

матрица A положительно определена [3], а добавление положительного коэффициента $\frac{\gamma}{\tau^2}$ к главной диагонали матрицы A только «усиливает» положительную определенность. Решение схемы (12) можно также получить из приращений $\Delta\varphi^{j+1}$. Для этого на каждом временном шаге надо решать на клеточной нейронной сети систему уравнений

$$\left(A + \frac{\gamma}{\tau^2} E \right) \Delta\varphi^{j+1} = F, \quad (13)$$

где $F = \frac{\gamma}{\tau^2} (2\Delta\varphi^j - \Delta\varphi^{j-1}) + \Delta f^j$,

А затем можно найти решение на очередном шаге (10).

Рассмотрим теперь абсолютно устойчивую схему Кранка-Николсона (7). Эту схему можно записать так

$$\left(A + \frac{2\gamma}{\tau^2} E \right) \varphi^{j+1} = \frac{4\gamma}{\tau^2} \varphi^j - \left(A + \frac{2\gamma}{\tau^2} E \right) \varphi^{j-1} + 2f^j \quad (14)$$

Матрица $\left(A + \frac{2\gamma}{\tau^2} E \right)$ положительно определена и поэтому система (14) может быть решена на клеточной нейронной сети. Для приращений можно записать

$$\left(A + \frac{2\gamma}{\tau^2} E \right) \Delta\varphi^{j+1} = F, \quad (15)$$

$$\text{где } F = \frac{4\gamma}{\tau^2} \Delta\varphi^j - \left(A + \frac{2\gamma}{\tau^2} E \right) \Delta\varphi^{j-1} + 2\Delta f^j.$$

Очередное решение находится по формуле (10).

Библиографический список

1. Тимошенко, С. П. Колебания в инженерном деле / С. П. Тимошенко, А. Х. Янг, У. Уивер. – М. : Машиностроение, 1985. – 472 с.
2. Самарский, А. А. Численные методы / А. А. Самарский, А. В. Гулин. – М. : Наука, Гл. ред. физ.-мат. лит., 1989. – 432 с.
3. Горбаченко, В. И. Нейрокомпьютеры в решении краевых задач теории поля / В. И. Горбаченко. – М. : Радиотехника, 2003. – 336 с.

С. В. Тумасов

Пензенский государственный университет, Пенза, Россия

РАСПОЗНАВАНИЕ РЕЧИ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ

Представлен обзор методов распознавания речи с ориентацией на методы определения ключевых слов (KWS) в сетях передачи данных, приведены особенности работы и возможности их реализации.

Определение ключевых слов (KWS) обеспечивает увеличение скорости передачи речевой информации в сети, передавая не все отсчёты речи целиком, а только важные ключевые слова в двоичном формате, которые она смогла распознать. При этом от методов определения ключевых слов требуется быстрая реакция в реальном масштабе времени и высокая точность [1].

В последнее время нейронные сети стали привлекательным выбором для архитектуры KWS из-за их превосходной точности по сравнению с традиционными алгоритмами обработки речи. Приложения KWS обычно работают на крошечных микроконтроллерах с ограниченной памятью и вычислительной способностью. Дизайн архитектуры нейронной сети для KWS должны учитывать эти ограничения.

В данной статье выполнен обзор различных архитектур нейронных сетей, применимых в KWS.

Потребление низкой потребляемой мощности для систем определения ключевых слов делает микроконтроллеры очевидным выбором для развертывания KWS в постоянно действующей системе.

Однако развертывание нейронная сеть KWS на микроконтроллерах имеет следующие проблемы:

1. Ограниченный объем памяти. Типичные системы микроконтроллеров имеют от десятков до нескольких сотен КБ доступной памяти. Вся модель нейронной сети, включая вход/выход, вес и активацию, должна соответствовать этому небольшому бюджету памяти.

2. Ограниченные вычислительные ресурсы. Поскольку KWS всегда включен, требования в режиме реального времени ограничивают общее количество операций на выводе нейронной сети.

Эти ограничения ресурсов микроконтроллера в сочетании с высокой точностью и низкой задержкой требования KWS требуют исследования нейронной сети с ограниченными ресурсами, чтобы найти ориентированных на KWS, что является основным направлением статьи.

Типичная система KWS состоит из экстрактора функций и классификатора на основе нейронной сети, как показано на рис. 1.

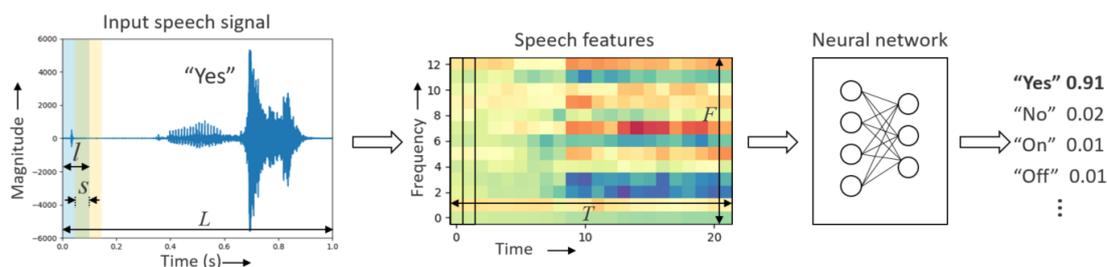


Рис. 1. Конвейер определения ключевого слова

Из каждого кадра функции речи F делается извлечение, генерирующее всего $T \times F$ признаков для всего входного речевого сигнала длины L . LFBE и MFCC используют человеческие инженерные речевые особенности в распознавании речи на основе глубокого обучения.

Выделение функций с использованием LFBE или MFCC включает перевод речевого сигнала во временной области в набор спектральных коэффициентов частотной области, который обеспечивает сжатие размерности входного сигнала. Выбранная матрица характеристик речи поступает в модуль классификатора, который генерирует вероятности для выходных классов. В реальном мире сценарий, в котором ключевые слова должны быть идентифицированы из непрерывного аудиопотока.

Традиционные технологии распознавания речи для KWS используют скрытые марковские модели (HMM) и декодирование Витерби. Хотя эти подходы достигают разумной точности, они сложны в тренировке и вычислительно дорожают во время вывода.

Другие методы исследования KWS включают дискриминационные модели, принимающие формулировку с большими запасами или рекуррентные нейронные сети (RNN). Хотя эти методы значительно превосходят KWS на основе HMM в терминах точности, они страдают от большой задержки обнаружения.

Модели KWS, использующие глубокие нейронные сети (DNN) основанные на полностью связанных слоях с выпрямленными линейными модулями (ReLU), активируются функцией активации, которая превосходит модели HMM с очень малой задержкой обнаружения. Основным недостатком DNN является то, что они игнорируют локальную временную и спектральную корреляцию во входных речевых функциях.

Чтобы использовать эти корреляции, изучаются различные варианты сверточных нейронных сетей (CNN) KWS, которые демонстрируют более высокую точность, чем DNN. Недостаток CNN в моделировании времени (например, речь) заключается в том, что они игнорируют долгосрочные временные зависимости.

Хотя все ранее существовавшие нейронные сети KWS тренируются с функцией потерь кросс-энтропии, функция потерь на основе максимального пула для обучения KWS-модель с длинной кратковременной памятью (LSTM) предлагается в [2], которая обеспечивает лучшую точность, чем DNN и LSTM, прошедшие обучение с потерей кросс-энтропии.

Свертка рекуррентной нейронной сети (CRNN) [3] представляет собой гибридную CNN и RNN, который использует преимущества и то и другое. Он использует локальную временную/пространственную корреляцию, используя уровни свертки и глобальные временные зависимости в речевых функциях с использованием повторяющихся слоев.

В последнее время в качестве эффективной альтернативы стандарту предложена 3-D операция свертки и была использована для создания компактных сетевых архитектур в области компьютерного зрения. DS-CNN сначала свертывает каждый канал на карте входных функций с отдельным 2-D фильтром и затем использует поточечные свертки (т.е. 1×1) для объединения выходов в глубину [4].

Таблица 1 суммирует точность, потребность в памяти и операции на вывод для сети архитектуры KWS, подготовленные по набору данных речевых команд Google.

Таблица 1

Потребности архитектур нейронных сетей в памяти и в вычислительных возможностях

| NN model | S(80KB, 6MOps) | | | M(200KB, 20MOps) | | | L(500KB, 80MOps) | | |
|------------|----------------|--------|--------|------------------|---------|--------|------------------|---------|--------|
| | Acc. | Mem. | Ops | Acc. | Mem. | Ops | Acc. | Mem. | Ops |
| DNN | 84.6% | 80.0KB | 158.8K | 86.4% | 199.4KB | 397.0K | 86.7% | 496.6KB | 990.2K |
| CNN | 91.6% | 79.0KB | 5.0M | 92.2% | 199.4KB | 17.3M | 92.7% | 497.8KB | 25.3M |
| Basic LSTM | 92.0% | 63.3KB | 5.9M | 93.0% | 196.5KB | 18.9M | 93.4% | 494.5KB | 47.9M |
| LSTM | 92.9% | 79.5KB | 3.9M | 93.9% | 198.6KB | 19.2M | 94.8% | 498.8KB | 48.4M |
| GRU | 93.5% | 78.8KB | 3.8M | 94.2% | 200.0KB | 19.2M | 94.7% | 499.7KB | 48.4M |
| CRNN | 94.0% | 79.7KB | 3.0M | 94.4% | 199.8KB | 7.6M | 95.0% | 499.5KB | 19.3M |
| DS-CNN | 94.4% | 38.6KB | 5.4M | 94.9% | 189.2KB | 19.8M | 95.4% | 497.6KB | 56.9M |

CRNN, которые сочетают в себе лучшие свойства CNN и RNN, обеспечивают лучшую точность, чем CNN и RNN, даже с меньшим количеством Ops. CRNN архитектура также масштабируется, когда доступно больше ресурсов памяти/вычислений. DS-CNN обеспечивает наилучшую точность и демонстрирует хорошую масштабируемость благодаря своей более глубокой архитектуре с помощью глубоких разделимых слоев свертки, которые менее интенсивно вычисляются/запоминаются.

Авторами в данной работе была подчеркнута роль KWS в сетях передачи данных. Аппаратная оптимизированная нейронная сетевая архитектура является ключом к получению эффективных результатов в памяти и вычислению.

Библиографический список

1. Таранцев, Е. К. Сеть радиомаяков с расширенным спектром / Е. К. Таранцев, С. В. Тумасов // XXI век: итоги прошлого и проблемы настоящего плюс. – Пенза : Изд-во ПензГТУ, 2016. – URL: <http://elibrary.ru>
2. Yundong, Zhang Hello Edge : Keyword Spotting on Microcontrollers, Arm / Zhang Yundong, Suda Naveen, Lai Liangzhen and Chandra Vikas. – San Jose, CA, 2018. – URL: <http://arxiv.org>
3. Max-pooling loss training of long short-term memory networks for small-footprint keyword spotting / Ming Sun, Anirudh Raju, George Tucker, Sankaran Panchapagesan, Gengshen Fu, Arindam Mandal, Spyros Matsoukas, Nikko Strom, and Shiv Vitaladevuni // In Spoken Language Technology Workshop (SLT), 2016 IEEE. – San Juan, 2016. – P. 474–480.
4. Convolutional recurrent neural networks for small-footprint keyword spotting / Sercan O Arik, Markus Kliegl, Rewon Child, Joel Hestness, Andrew Gibiansky, Chris Fougner, Ryan Prenger, and Adam Coates. – arXiv preprint arXiv:1703.05390, 2017.

УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ, ЭКОНОМИЧЕСКИХ И ТЕХНИЧЕСКИХ СИСТЕМАХ

О. О. Ашкирина, З. И. Баусова, М. И. Гринцов, Н. В. Дорогова

Пензенский государственный университет, Пенза, Россия

ИСПОЛЬЗОВАНИЕ БИНАРНОЙ ЛОГИСТИЧЕСКОЙ МОДЕЛИ В МЕДИЦИНСКИХ ИССЛЕДОВАНИЯХ

Статистические методы обработки данных могут быть применимы во многих сферах деятельности человека. Они позволяют с помощью математического моделирования с использованием исходных данных получить информацию о влиянии параметров на результирующий показатель решаемой задачи.

Логистическая регрессия – это разновидность множественной регрессии, назначение которой состоит в анализе связи между несколькими независимыми переменными и зависимой переменной. С помощью бинарной логистической регрессии можно оценивать вероятность того, что событие наступит для конкретного испытуемого (больной/здоровый). Поэтому данная модель применима на начальных этапах исследования, когда необходимо принять или опровергнуть гипотезу.

В современном мире, когда приходится обрабатывать огромное количество информации в условиях ограниченного времени, преимущество отдается компьютерным программам для статистической обработки данных.

Одним из лидеров рынка в области статистических продуктов, предназначенных для проведения прикладных исследований, является компания IBM SPSS и ее программный продукт SPSS Statistics. Программное обеспечение IBM SPSS предлагает множество вариантов обобщения и подытоживания данных, выполнения статистических проверок, а также реализацию большого числа методов моделирования (как для прогностического анализа, так и для выявления групп и ассоциаций) и обширный набор методов визуализации данных. Программные платформы IBM SPSS поддерживают автоматизацию аналитических решений любого уровня таким образом, чтобы аналитика стала неотъемлемым элементом принятия повседневных решений для специалистов различных отраслей./1/

Система SPSS позволяет решать задачи с бинарным откликом с помощью логистической регрессии. Ввод исходных данных выполняется аналогично множественной линейной регрессии.

В рамках данного исследования предполагается проверить гипотезу доктора медицинских наук М. И. Гринцова, согласно которой существует непосредственное влияние противостояния планет (оппозиции планет) на даты зачатия и рождения детей с врожденными патологическими заболеваниями или их отсутствием.

Исходными данными для проверки выдвинутой гипотезы является информация о больных и здоровых детях, а именно:

- даты зачатия и рождения;
- наличие оппозиции планет на даты зачатия и рождения;
- наличие/отсутствие болезни (1 – больной ребенок, 0 – здоровый).

Фрагмент исходной информации представлен на рис. 1.

Для анализа исходных данных выбрана бинарная логистическая модель с функцией распределения, в которой в качестве аргумента x используется линейная комбинация независимых переменных:

$$F(x) = \frac{e^x}{1 + e^x}$$

В качестве независимых переменных были объявлены следующие: дата зачатия $x1$, противостояние на дату зачатия $x2$, дата рождения $x3$, противостояние на дату рождения $x4$. Зависимой переменной Y выбран показатель «Наличие болезни» (0 или 1).

В качестве методов построения бинарной логистической модели были использованы:

– принудительное включение (Enter) – процедура отбора, при которой все переменные вводятся за один шаг;

– отбор исключением – шаговый отбор переменных.

Результаты моделирования, полученные вышеперечисленными методами, представлены на рис. 2–6.

| Дата зачатия | Дата зачатия | Противостояние на дату зачатия | Дата рождения | Дата рождения | Противостояние на дату рождения | Наличие болезни |
|--------------|--------------|--------------------------------|---------------|---------------|---------------------------------|-----------------|
| 01.08.1938 | 213 | 1 | 01.05.1939 | 121 | 1 | 0 |
| 16.04.1952 | 107 | 1 | 14.01.1953 | 14 | 1 | 0 |
| 03.11.1966 | 307 | 1 | 04.08.1967 | 216 | 1 | 1 |
| 05.09.2009 | 247 | 1 | 05.06.2010 | 156 | 1 | 1 |
| 08.09.1964 | 252 | 1 | 08.06.1965 | 159 | 1 | 0 |
| 20.07.1970 | 201 | 1 | 19.04.1971 | 109 | 1 | 0 |
| 08.08.2013 | 220 | 1 | 06.05.2014 | 126 | 1 | 1 |
| 08.03.1981 | 67 | 1 | 06.12.1981 | 340 | 1 | 0 |
| 10.07.2008 | 191 | 1 | 09.04.2009 | 99 | 1 | 1 |
| 10.11.2003 | 314 | 1 | 09.08.2004 | 222 | 1 | 1 |

Рис. 1. Фрагмент массива исходной информации

Таблица классификации^а

| Наблюдаемые | | Predicted | | |
|-------------|-----------------|-----------------|-----|-----------------------|
| | | Наличие болезни | | Процент корректных |
| | | ,0 | 1,0 | |
| Шаг 1 | Наличие болезни | ,0 | 1,0 | 53,7 |
| | | 22 | 19 | |
| | Общий процент | 8 | 52 | 86,7 |
| | | | | 73,3 |

а. Разделяющее значение = ,500

Рис. 2. Таблица классификации (метод Enter)

Переменные в уравнении

| | | В | Стд.Ошибка | Вальд | ст.св. | Знч. | Exp(B) |
|--------------------|------------------------------|--------|------------|-------|--------|------|--------|
| Шаг 1 ^а | Датазачатия_А | -,003 | ,002 | 1,072 | 1 | ,300 | ,997 |
| | Противостояниенадатузачатия | 1,931 | ,638 | 9,159 | 1 | ,002 | 6,897 |
| | Датарождения_А | ,000 | ,002 | ,039 | 1 | ,844 | 1,000 |
| | Противостояниенадатурождения | 1,379 | ,539 | 6,551 | 1 | ,010 | 3,973 |
| | Константа | -1,635 | ,893 | 3,355 | 1 | ,067 | ,195 |

а. Переменные, включенные на шаге 1: Датазачатия_А, Противостояниенадатузачатия,

Рис. 3. Итоговая таблица (метод Enter)

По таблице рис. 2 следует, что качество модели (метод Enter), т.е. процент корректно предсказанных событий, равен 73 %, а по методу «Отбор исключением» (рис. 4) – 70 %. Так как процент корректно предсказанных событий по методу Enter больше, чем по методу «Отбор исключением», следовательно, качество модели по методу Enter лучше.

Значимым параметром является тот параметр, уровень значимости которого меньше 0,05. Анализируя результаты таблицы на рис. 3, значимыми параметрами являются противостояние на дату зачатия, уровень значимости которого равен 0,002 и противостояние на дату рождения, уровень значимости которого равен 0,01.

Таблица классификации^а

| Наблюдаемые | | Predicted | | | |
|-------------|-----------------|-----------------|-----|-----------------------|------|
| | | Наличие болезни | | Процент корректных | |
| | | ,0 | 1,0 | | |
| Шаг 1 | Наличие болезни | ,0 | 22 | 19 | 53,7 |
| | | 1,0 | 8 | 52 | 86,7 |
| | Общий процент | | | | 73,3 |
| Шаг 2 | Наличие болезни | ,0 | 23 | 18 | 56,1 |
| | | 1,0 | 11 | 49 | 81,7 |
| | Общий процент | | | | 71,3 |
| Шаг 3 | Наличие болезни | ,0 | 25 | 16 | 61,0 |
| | | 1,0 | 14 | 46 | 76,7 |
| | Общий процент | | | | 70,3 |

а. Разделяющее значение = ,500

Рис. 4. Таблица классификации (метод «Отбор исключением»)

Переменные в уравнении

| | В | Стд.Ошибка | Вальд | ст.св. | Знч. | Exp(В) | |
|--------------------|-------------------------------|------------|-------|--------|------|--------|-------|
| Шаг 1 ^а | | | | | | | |
| | Датазачатия_А | -,003 | ,002 | 1,072 | 1 | ,300 | ,997 |
| | Противостояниенадатуз ачатия | 1,931 | ,638 | 9,159 | 1 | ,002 | 6,897 |
| | Датарождения_А | ,000 | ,002 | ,039 | 1 | ,844 | 1,000 |
| | Противостояниенадатур ождения | 1,379 | ,539 | 6,551 | 1 | ,010 | 3,973 |
| | Константа | -1,635 | ,893 | 3,355 | 1 | ,067 | ,195 |
| Шаг 2 ^а | | | | | | | |
| | Датазачатия_А | -,002 | ,002 | 1,032 | 1 | ,310 | ,998 |
| | Противостояниенадатуз ачатия | 1,942 | ,636 | 9,320 | 1 | ,002 | 6,970 |
| | Противостояниенадатур ождения | 1,357 | ,526 | 6,655 | 1 | ,010 | 3,885 |
| | Константа | -1,715 | ,798 | 4,623 | 1 | ,032 | ,180 |
| Шаг 3 ^а | | | | | | | |
| | Противостояниенадатуз ачатия | 2,081 | ,625 | 11,088 | 1 | ,001 | 8,011 |
| | Противостояниенадатур ождения | 1,224 | ,503 | 5,923 | 1 | ,015 | 3,402 |
| | Константа | -2,170 | ,680 | 10,178 | 1 | ,001 | ,114 |

а. Переменные, включенные на шаге 1: Датазачатия_А, Противостояниенадатузачатия, Датарождения_А, Противостояниенадатурождения.

Рис. 5. Итоговая таблица (метод «Отбор исключением»)

Фрагмент результатов прогноза рождения больных детей по обоим методом представлен на рис. 6.

| | Датазачатия | Датазачатия_А | Противостояниена датузачатия | Датарождения | Датарождения_А | Противостояниена датурождения | Наличиеболезни | PRE_1 | PRE_2 |
|-----|-------------|---------------|------------------------------|--------------|----------------|-------------------------------|----------------|---------|---------|
| 91 | 37766 | 144,0 | 1,0 | 38039 | 53,0 | 0,0 | 1,0 | 0,47777 | 0,47710 |
| 92 | 41083 | 175,0 | 1,0 | 41356 | 82,0 | 0,0 | 1,0 | 0,47777 | 0,45443 |
| 93 | 14.01.1814 | 14,0 | 0,0 | 14.10.1814 | 287,0 | 0,0 | 0,0 | 0,10249 | 0,14228 |
| 94 | 39585 | 137,0 | 1,0 | 39858 | 45,0 | 1,0 | 1,0 | 0,75681 | 0,78735 |
| 95 | 30625 | 309,0 | 0,0 | 30898 | 217,0 | 0,0 | 0,0 | 0,10249 | 0,07503 |
| 96 | 30701 | 20,0 | 0,0 | 30974 | 293,0 | 0,0 | 0,0 | 0,10249 | 0,14012 |
| 97 | 38058 | 72,0 | 1,0 | 38331 | 345,0 | 0,0 | 1,0 | 0,47777 | 0,49063 |
| 98 | 41020 | 112,0 | 1,0 | 41288 | 14,0 | 0,0 | 1,0 | 0,47777 | 0,50158 |
| 99 | 40194 | 16,0 | 1,0 | 40466 | 288,0 | 0,0 | 1,0 | 0,47777 | 0,53223 |
| 100 | 41019 | 111,0 | 1,0 | 41292 | 18,0 | 0,0 | 1,0 | 0,47777 | 0,50178 |
| 101 | 35925 | 129,0 | 0,0 | 36198 | 38,0 | 0,0 | 0,0 | 0,10249 | 0,12150 |

Рис. 6. Фрагмент сводной таблицы результатов ретропрогноза обоим методом, где PRE_1 – метод «отбор исключением»; PRE_2 – метод «Enter»

Уравнение регрессии выглядит следующим образом:

$$Y = -1,635 - 0,003x1 + 1,931x2 + 1,379x4$$

Анализ полученных результатов моделирования позволяет сделать вывод о том, что противостояние планет влияет на дату зачатия ребенка, вследствие чего ребенок родится с врожденной патологией.

Библиографический список

1. Многомерный статистический анализ в экономических задачах: компьютерное моделирование в SPSS : учеб. пособие / под ред. И. В. Орловой. – М. : Вузовский учебник, 2009. – 310 с.

А. К. Горелова

Пензенский государственный университет, Пенза, Россия

АНАЛИЗ ВОЗМОЖНОСТЕЙ МОБИЛЬНЫХ СИСТЕМ ЭЛЕКТРОННОГО ЗДРАВООХРАНЕНИЯ

Представлены возможности мобильного здравоохранения, помощь в хранении и обработке информации, своевременное получение консультаций у врача.

Благодаря появляющимся новым технологиям медицинской диагностики, сегодня можно более эффективно и качественно проводить исследования всего организма человека на предмет определения заболеваний и их причин, образования различных патологий.

По данным Всемирной организации здравоохранения, проблемы с сердцем являются одними из основных причин смертности населения. В год в мире от сердечно-сосудистых заболеваний умирает 17,5 млн человек. Именно поэтому одна из актуальных задач российского здравоохранения заключается в снижении смертности от сердечно-сосудистых заболеваний. В нашей стране, взрослое поколение пренебрегает ежегодным комплексным осмотрам и обращается к врачам, когда уже начались серьезные проблемы со здоровьем. До 35 лет проблемы с сердцем можно поправить коррекцией образа жизни и витаминами, если находиться под наблюдением врача.

Один из самых простых, доступных и информативных методов диагностики заболеваний сердца – электрокардиография (ЭКГ). Он основан на регистрации электрических импульсов, возникающих в сердце, и их графической записи в виде зубцов на специальную бумажную пленку. На основании этих данных можно судить не только об электрической активности сердца, но и о структуре миокарда. Это значит, что с помощью ЭКГ можно диагностировать множество различных заболеваний сердца [1].

Самостоятельная расшифровка ЭКГ человеком, не имеющим специальных медицинских познаний, невозможна. Все что может простой человек – лишь ориентировочно оценить отдельные параметры электрокардиограммы, соответствуют ли они норме и о какой патологии могут говорить. Но окончательные выводы по заключению сделать лишь квалифицированный специалист – врач-кардиолог, а также терапевт или семейный врач. Однако не у каждого есть время периодически ездить в клинику, особенно из удаленных населенных пунктов, поэтому в США давно практикуется дистанционное наблюдение, когда пациент делает нужные замеры дома и высылает результаты своему врачу. Не так давно на Российском рынке появились аналогичные портативные устройства. Примерами таких устройств являются кардиокварк, ритмер, кардиофлэшка.

Система кардиокварк (CardioQVARK) – с виду кардиомонитор напоминает чехол-аккумулятор. На задней крышке кардиомонитора располагаются два датчика, сверху вставляется телефон, и с помощью порта осуществляется обмен данными. Для снятия своей кардиограммы, необходимо скачать приложение, заполнить свои данные, включая возраст,

вес, вредные привычки, принять удобное положение тела, приложить два пальца к датчикам и постараться в течение 5 мин не двигаться. После окончания измерений данные отправляются на сервер для расшифровки и через несколько секунд появляются в карточке пациента.

Ритмер (ritmer) или теле-холтер – это персональный кардиорегистратор с функцией автоматического анализа ЭКГ для дистанционного и самостоятельного мониторинга. На тело человека устанавливается аппарат – портативный рекордер, который проводит постоянный мониторинг состояния сердца и его деятельности. Оборудование ведет регистрацию кардиограммы, которая обладает большей информативностью, чем при обычной процедуре снятия показаний сердечной деятельности. Сам прибор крепится на груди с помощью клеящихся электродов и регистрирует сигналы сердца. Приложение, через которое осуществляется анализ мониторинга ЭКГ, выдает автоматическую рекомендацию: когда нужно отдохнуть, когда увеличить нагрузку, а когда – обратиться к врачу. Можно ежедневно наблюдать за своим состоянием и оценивать динамику показателей. Также за этими данными могут следить и близкие люди в удаленном режиме.

Кардиофлэшка – позволяет проводить экспресс-мониторинг работы сердца. Поскольку у кардиофлэшки отсутствует и собственное хранилище данных, для регистрации измерений этот гаджет необходимо соединить со смартфоном или планшетом. Как известно, она позволяет подключать к порту microUSB различные периферийные устройства. Тремя составными частями кардиокомплекса являются: кардиофлэшка, соответствующее мобильное приложение и веб-сервис «облачное хранилище». Этот комплекс, в частности, позволяет измерять частоту сердечных сокращений, просматривать электрокардиограмму в реальном времени, а также сохранять результаты измерений в файле для отправки их удаленному специалисту (или распечатки, чтобы показать лечащему врачу в поликлинике).

Данные устройства имеют ряд преимуществ:

- сквозная система контроля и мониторинга здоровья;
- комплексная диагностика широкого спектра заболеваний;
- ранее выявление развития патологий;
- мгновенное реагирование на критические изменения здоровья;
- позволяет проводить исследования сердца всем, независимо от пола и возраста;
- сокращает затраты на медицинское обслуживание;
- при использовании не требует медицинского и технического образования;
- прост в эксплуатации.

Главная особенность всех устройств – внешний мобильный ЭКГ-монитор с возможностью удаленного контроля за состоянием пациента. Возможности, которые дают эти устройства, особенно актуальны для оказания срочной кардиологической помощи или врачебной консультации для человека в период обострения [2].

На рис. 1 представлена схема взаимодействия пациента и медицинского работника с помощью мобильной системы электронного здравоохранения с использованием мобильного устройства контроля работы сердца [3].

Предполагается наличие единой информационной СУБД, где будет храниться история болезни пациентов, так называемая «электронная амбулаторная карта». Карта содержит всю диагностическую информацию, относящуюся к конкретному пациенту. Информационный блок содержит заключение по данному исследованию и исходные кривые. Карта пациента позволяет оценить клиническую картину в целом, проследить динамику, включая влияние функциональных проб, соотнести между собой результаты различных исследований. В электронной амбулаторной карте будет предусмотрена возможность повторно проанализировать запись, или выполнить новое исследование. При этом предусматривается возможность анализа кардиографических данных с помощью специальных программных средств, например [4], что позволит значительно повысить качество и эффективность диагностики.

Квалифицированные специалисты медицинских учреждений будут иметь доступ для просмотра, редактирования и удаления той или иной информации о пациентах в зависимости от роли, предоставленной им в автоматизированной системе. Для получения доступа пациенту

к своей электронной амбулаторной карте, необходимо будет зарегистрироваться на специализированном сайте, и в дальнейшем для работы с ней осуществлять вход через личный кабинет. Для отправки результатов с измерительных приборов, таких как кардиограф, тонометр и др., необходимо будет через личный кабинет отправлять данные в электронную информационную СУБД, где полученная информация от пациентов будет автоматически фиксироваться и записываться в электронную амбулаторную карту. Задача врачей – отслеживать состояние пациентов и вовремя давать рекомендации (см. рис. 1).

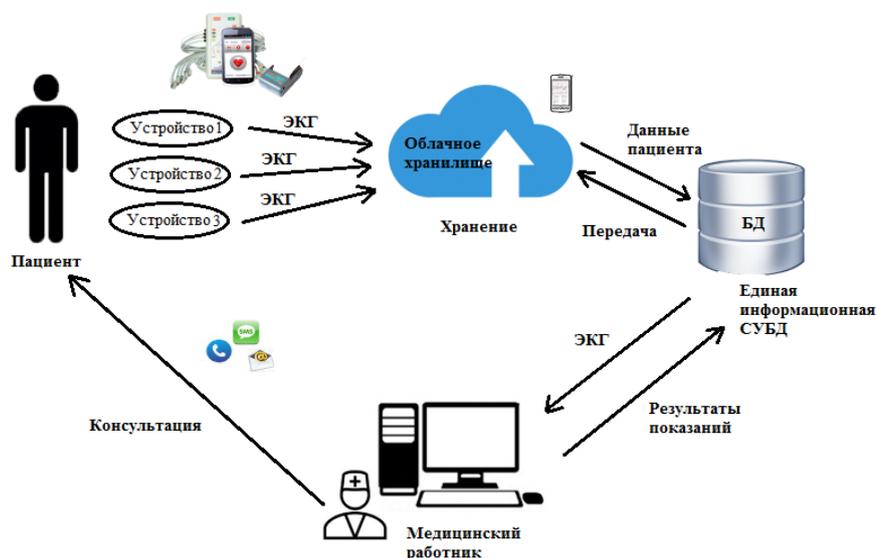


Рис. 1 . Схема работы мобильной системы электронного здравоохранения

К преимуществам данной системы следует отнести: во-первых, если вся информация о пациенте будет храниться в электронном виде в одной общей базе данных, то врач, находясь в любом городе или даже стране, сможет быстро получить нужную ему информацию, соответственно уменьшается риск утери какой-либо информации. Во-вторых, современные технологии позволяют провести консультацию у любого врача, не выходя из дома. Это очень удобно для пациентов, живущих далеко от больницы, пожилых людей, а также людей с ограниченными возможностями. Следует также отметить, что у этой системы очень высокая степень защиты, и вероятность того, что какая-либо информация попадет к посторонним людям, минимальна.

Мобильное здравоохранение (mHealth) активно развивается, уже разработаны и используются приложения, которые помогают людям, выявить симптомы депрессии, контролировать количество потребляемых калорий, напоминают о приеме необходимых лекарств и т.д. Единая информационная система электронного здравоохранения поможет хранить и обрабатывать информацию в одном месте, при этом пациент сможет своевременно получить консультацию у врача и не заниматься самолечением, так как это может только усугубить ситуацию.

Библиографический список

1. Калиниченко, А. Н. Алгоритмическое обеспечение анализа ЭКГ системы дистанционного кардиологического мониторинга / А. Н. Калиниченко, В. Л. Емалов. – СПб. , 2018. – С. 92.
2. Куракова, Н. Информатизация здравоохранения как инструмент создания «саморегулируемой системы организации медицинской помощи» / Н. Куракова // Врач и информационные технологии. – 2009. – № 2. – С. 4–27.
3. Mobile heart monitoring system prototype based on the Texas Instruments hardware: energy efficiency and J-point detection / A. Kuzmin, M. Safronov, O. Bodin, M. Petrovsky, A. Sergeenkov // International Journal of Embedded and Real-Time Communication Systems. – 2016. – Vol. 7, № 1. – P. 64–84.
4. Пат. 2257838 RUS. Способ диагностики состояния сердечно-сосудистой системы / Бодин О. Н., Агапов Е. Г., Адамов А. В., Бурукина И. П., Кузьмин А. В. (от 09.03.2004).

З. И. Баусова, А. Ю. Старикова, А. Н. Гамазина, Ю. Н. Дугина

Пензенский государственный университет, Пенза, Россия

АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Представлен анализ методов решения задач нелинейного программирования. Нелинейное программирование – это раздел математического программирования, который объединяет теорию и методы решения задач отыскания экстремальных значений, в которых целевая функция содержит нелинейные выражения [1].

Задачами выпуклого программирования называют задачи нелинейного программирования, где все функции f, f_1, \dots, f_m являются выпуклыми [2]. В решении задач нелинейного программирования нет какого-либо определённого алгоритма. Выпуклое (нелинейное) программирование включает в себя методы решения, которые описываются нелинейными соотношениями. Нелинейное программирование применяется в таких областях, как управление товарными ресурсами, промышленное производство и прогнозирование рынка и т.д.

Решение таких задач производится большим количеством методов, среди которых метод кусочно-линейной аппроксимации и метод спуска.

Рассмотрим суть метода спуска и наискорейшего спуска. Градиентом функции $F(x)$ называется вектор, координаты которого являются частными производными этой функции. В каждой точке X направление градиента является направлением наибольшего возрастания функции.

Основной алгоритм решения задач методами спуска заключается в построении последовательности X_0, X_1, \dots, X_k решений системы ограничений задачи. В значение X_0 выбирается любая точка области решений и каждая последующая точка вычисляется по следующей формуле:

$$X_{k+1} = X_k + \lambda l \quad (1.1)$$

где l – это направление вектора, а λ – число. l и λ выбираются так, чтобы обеспечить сходимость последовательности к оптимальному решению. Когда процесс получения последовательных приближений X_k бесконечен, тогда X_{k_0} берется за приближенное значение оптимального решения. Но процесс также может завершиться и за конечное число шагов.

Методы спуска, в которых последовательности находятся по следующим формулам, являются градиентными:

$$X_{k+1} = X_k + \lambda * \Delta Z(X_k), \lambda > 0 \text{ (если ищется } Z_{\max}) \quad (1.2)$$

$$X_{k+1} = X_k - \lambda * \Delta Z(X_k), \lambda > 0 \text{ (если ищется } Z_{\min}) \quad (1.3)$$

Разница между ними заключается в способах выбора длины шага λ и схемами нахождения точки X_{k+1} .

Далее рассмотрим суть метода наискорейшего спуска.

По этому методу длина шага λ в формулах (1.2) и (1.3) выбирается так, чтобы λ достигался экстремум функции:

$$\Delta Z = Z(X_{k-1}) - Z(X_k) \quad (1.4)$$

Выбор λ осуществляется так, чтобы приращение функции ΔZ при перемещении их точки X_k в точку X_{k+1} было наибольшим, для поиска Z_{\max} , или наименьшим при поиске Z_{\min} . Необходимым условием экстремума, если использовать скалярное произведение векторов:

$$\Delta Z(X_{k+1}) * \Delta Z(X_k) = 0. \quad (1.5)$$

В том случае, если оптимум находится внутри области решений системы ограничений заданной задачи, то точка X_{k+1} , найденная по формуле (1.2) или (1.3) не выйдет за пределы этой области.

В случае, когда оптимум целевой функции находится на границе области решений системы ограничений, также в качестве точки X_0 взяв любую точку из области решений и

находя последующие точки по формулам (1.2) и (1.3), мы получим то, что точка X_k уже не лежит в области решений. Нужно вместо точки X_k взять точку, которая лежит на пересечении направления спуска с границей области решений. Тогда оптимум достигается в той точке, в которой градиент перпендикулярен границе области решений.

Другим методом приближенного решения задач выпуклого программирования является метод кусочно-линейной аппроксимации.

Допустим, что в задаче выпуклого программирования функция цели Z и все ограничения φ_i – сепарабельные (это такие функции, которые можно представить в виде суммы функций, каждая из которых зависит только от одной переменной). При этом задача примет вид: найти минимум выпуклой (максимум – вогнутой) функции $Z = \sum_{j=1}^n f_j(x_j)$ при таких

ограничениях, как:

$$\sum_{j=1}^n \varphi_{ij}(x_j) \leq b_i, \quad i = 1, \dots, m. \quad (1.6)$$

Метод кусочно-линейной аппроксимации заключается в том, что происходит замена f_i и φ_{ij} на ломанные линии, которые состоят из прямолинейных отрезков. При всём этом задача ВП которая была в начале, заменяется новой, приближённой, которая является задачей линейного программирования. Обычно такие задачи решаются симплексным методом и их решение является приближенным решением исходной задачи выпуклого программирования [3].

Для построения приближенной задачи рассмотрим кусочно-линейную аппроксимацию функций одной переменной $h(x)$, которая задана на отрезке $[0, a]$. Нам необходимо разбить отрезок на g частей точками $x_0 < x_1 < \dots < x_r$ так, чтобы $x_0 = 0, x_r = a$ (рис. 1.1).

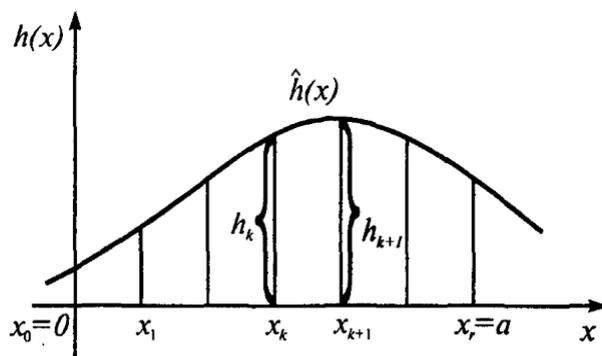


Рис. 1.1

Далее нужно вычислить значения функции $h_k(x)$ ($k = 0, \dots, r$) в этих точках и соединить попарно точки $(x_k; h_k)$ и $(x_{k+1}; h_{k+1})$ отрезками прямых. Состоящая из таких отрезков ломаная $\hat{h}(x)$ аппроксимирует функцию $h(x)$ на отрезке $[0, a]$. (Аппроксимация – научный метод, состоящий в замене одних объектов другими, в каком-то смысле близкими к исходным, но более простыми). В таком случае, мы не рассматриваем оценку точности приближения, но можно отметить, что её можно увеличить за счёт более мелкого разбиения отрезка.

Уравнение участка ломаной $\hat{h}(x)$ между точками $(x_k; h_k)$ и $(x_{k+1}; h_{k+1})$ будет иметь вид уравнения прямой по двум точкам, а если в данном выражении каждое соотношение обозначить через λ , то получим:

$$\begin{cases} x = \lambda x_{k+1} + (1 - \lambda)x_k, \\ h(x) = \lambda h_{k+1} + (1 - \lambda)h_k, \end{cases} \quad (1.7)$$

причём $0 \leq \lambda \leq 1$.

Так же нужно отметить, что для каждого $x \in [x_k, x_{k+1}]$ существует единственное значение λ , удовлетворяющее условиям (1.7). Если обозначить $1 - \lambda = \lambda_k, \lambda = \lambda_{k+1}$, то возможно переписать это уравнение в виде:

$$\begin{cases} x = \lambda_k x_k + \lambda_{k+1} x_{k+1}, \\ h(x) = \lambda_k h_k + \lambda_{k+1} h_{k+1}, \end{cases} \quad (1.8)$$

где $\lambda_k + \lambda_{k+1} = 1$, $\lambda_k \geq 0$, $\lambda_{k+1} \geq 0$.

Такие уравнения являются параметрическими уравнениями отрезка. Если $h(x) = 0$, то второе из этих уравнений обращается в тождество $0 = 0$, а первое из них будет принимать вид уравнения отрезка, лежащего на оси абсцисс (1.9).

$$x = \alpha_1 a + \alpha_2 b, \quad \alpha_1 + \alpha_2 = 1, \quad \alpha_1 \geq 0, \quad \alpha_2 \geq 0. \quad (1.9)$$

Отсюда, для любого $x \in [0, a]$ уравнение ломаной можно записать:

$$\begin{cases} x = \sum_{k=0}^r \lambda_k x_k, \\ h(x) = \sum_{k=0}^r \lambda_k h_k, \quad \sum_{k=0}^r \lambda_k = 1 \text{ и } \lambda_k \geq 0 \quad (k=0, \dots, r), \end{cases} \quad (1.10)$$

причём являются отличными от нуля только 2 значения λ_k (только если x является внутренней точкой k -го отрезка разбиения) либо только 1 значение (если x совпадает с концом отрезка).

Если заново рассматривать задачу ВП с сепарабельными функциями, можно отметить, что для начала нужно определить интервал изменений переменных x_j . После чего каждый интервал разбить на части точками x_{jk} и с использованием формул (1.10) построить кусочно-линейную аппроксимацию для функций f_j и φ_{ij} . Затем, для исходной задачи (1.6) записать приближенную задачу:

$$\begin{aligned} &\text{максимума функции } \hat{Z} = \sum_{j=1}^n \hat{f}_j(x_j) \\ &\text{при ограничениях } \sum_{j=1}^n \hat{\varphi}_{ij}(x_j) \leq b_i, \quad i = 1, \dots, m \\ & \quad \quad \quad x_j \geq 0, \quad j = 1, \dots, n \end{aligned} \quad (1.11)$$

Так как приближенную задачу (1.11) нужно решать симплексным методом, условия неотрицательности переменных нужно записывать отдельно от других ограничений. У таких задач есть некоторые отличия от обычных задач линейного программирования и они состоят в том, что для каждого x_j есть не более двух соседних ненулевых λ_{jk} и, получается, что нельзя брать основными переменными два λ_{jk} с одинаковым j и несоседними k . Так же нужно заметить, что для неотрицательных условий переменных слагаемых $f_j(x_j)$ и $\varphi_{ij}(x_j)$ (если такие существуют) кусочно-линейная аппроксимация не желательна.

Достоинством метода является то, что улучшить точность приближения возможно путём разбиения на более мелкие части переменных, которые взяты в окрестности полученного первого приближения. Важным недостатком является большое увеличение объема задачи (в том числе числа переменных) при переходе к приближенной линейной модели [4].

Таким образом, выбирая из рассматриваемых нами методов приближенного решения задач выпуклого программирования более оптимальным для использования являются методы спуска, так как они являются более эффективным для решения таких видов задач.

Библиографический список

1. Красс, М. С. Математические методы и модели для магистрантов экономики : учеб. пособие / М. С. Красс, Б. П. Чупрынов. – 2-е изд., доп. – СПб. : Питер, 2010. – 496 с.
2. Кремер, Н. Ш. Исследование операций в экономике : учеб. пособие для вузов / Н. Ш. Кремер, Б. А. Путко, И. М. Тришин, М. Н. Фридман ; под ред. проф. Н. Ш. Кремера. – М., 2005. – URL: <http://edu-lib.com>
3. Решение экономических задач с помощью математического моделирования : метод. указания к курсовой работе по дисциплине «Методы оптимизации» / сост.: З. И. Баусова, О. А. Голышевский, А. В. Еременко. – Пенза : Изд-во ПГУ, 2002. – 52 с.
4. Баусова, З. И. Исследование математических моделей экономических систем с применением теории оптимального управления : учеб. пособие с грифом УМО / З. И. Баусова, Н. В. Уткина, И. И. Шукшина ; под ред. А. П. Ремонтова. – Пенза : Изд-во ПГУ, 2010. – 163 с.

Пензенский государственный университет, Пенза, Россия

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ МАСШТАБИРОВАНИЯ И УПРАВЛЕНИЯ КОНТЕЙНЕРИЗИРОВАННЫМИ ПРИЛОЖЕНИЯМИ ДЛЯ РАСПРЕДЕЛЕННЫХ КЛАСТЕРНЫХ СИСТЕМ

Рассматриваются способы создания высоконагруженных распределенных кластерных приложений. Отмечается возможность использования веб-технологий для построения многокомпонентных систем. Описываются типичные составные части веб-приложения. Выявляются основные проблемы, связанные с обеспечением организации взаимодействия компонентов распределенной системы.

В настоящее время большую популярность приобретают программные решения, связанные с распределенными веб-технологиями и сервисами, предоставляющими либо единый интерфейс доступа к данным для множества клиентских приложений, либо законченную систему, совмещающую в себе хранение данных и реализацию интерфейсов. Большинство современных веб-приложений используют несколько серверов для повышения производительности и отказоустойчивости системы, объединенных в кластер. При плохом уровне масштабируемости системы с ростом машин процессы планирования, обслуживания, мониторинга и управления веб-приложением становятся трудоемкой задачей. Также сложность представляют процессы организации взаимодействия между различными частями приложения, автоматизации процессов обновления версий программного обеспечения, управления и мониторинга ресурсов и другие.

Типовое веб-приложение, как правило, состоит из следующих частей (сущностей):

- Сущность, отвечающая за хранение данных и управление ими. В этой роли выступают сервера базы данных. Они должны соблюдать требования по масштабируемости (способны задействовать дополнительные мощности для серверов СУБД, реплик и т.п.), надежности, безопасности.
- Сущность, отвечающая за обработку данных и содержащая бизнес-логику приложения. Они могут быть разбиты на подсистемы и физически разделены для повышения эффективности обработки.
- Сущность, отвечающая за взаимодействие с пользователем или с другими приложениями, представляющая интерфейс доступа к веб-приложению.

Для решения задач по мониторингу, управлению и поддержке многокомпонентных веб-приложений целесообразно использовать готовые решения, позволяющие автоматизировать процессы поддержки кластерной архитектуры системы. Одним из таких решений является фреймворк Kubernetes [1, с. 19]. Данное программное обеспечение бесплатное, с открытым исходным кодом, используется многими современными веб-приложениями и предназначено для управления кластером контейнеров Linux как единой системой. Данный фреймворк использует концепцию Docker контейнеров, что позволяет управлять и запускать контейнеры Docker на большом количестве хостов, а также обеспечивает совместное размещение и репликацию большого количества контейнеров. Проект был начат Google и теперь поддерживается многими компаниями, среди которых Microsoft, RedHat, IBM и Docker. Kubernetes представляет высокоуровневый API, определяющее логическое группирование контейнеров, позволяющее определять пулы контейнеров, балансировать нагрузку, а также задавать их размещение [2, с. 842].

Основные понятия и концепции Kubernetes:

- Node – это машина в кластере Kubernetes.
- Pod – это группа контейнеров с общими разделами, запускаемых как единое целое.

- ReplicationController – сущность, гарантирующая, что определенное количество реплик pod будут запущены в любой момент времени.
- Services – это абстракция, которая определяет логический объединенный набор pod и политику доступа к ним.
- Volumes – это директория, возможно, с данными в ней, которая доступна в контейнере.
- Labels – это пары ключ/значение которые прикрепляются к объектам, например к pod. Label могут быть использованы для создания и выбора наборов объектов.
- KubectlCommandLineInterface – интерфейс командной строки для управления Kubernetes.

Кластер Kubernetes включает в себя агента, запущенного на нодах (kubelet) и компоненты мастера (APIs, scheduler, etc), поверх решения с распределенным хранилищем. Архитектура системы упрощенно представлена в виде сервисов, которые работают на каждой ноде и сервисов уровня управления кластера. На каждой ноде Kubernetes запускаются сервисы, необходимые для управления нодой со стороны мастера и для запуска приложений. Конечно, на каждой ноде запускается Docker, который обеспечивает загрузку образов и запуск контейнеров. Kubelet управляет pods, их контейнерами, образами, разделами, и т. п. Также на каждой ноде запускается простой прокси-балансировщик (Kube-Proxy). Этот сервис запускается на каждой ноде и настраивается в Kubernetes API. Kube-Proxy может выполнять простейшее перенаправление потоков TCP и UDP (roundrobin) между набором бэкендов.

Состояние мастера хранится в экземпляре etcd. Это обеспечивает надёжное хранение конфигурационных данных и своевременное оповещение прочих компонентов об изменении состояния. Kubernetes API обеспечивает работу api-сервера. Он предназначен для того, чтобы быть CRUD сервером со встроенной бизнес-логикой, реализованной в отдельных компонентах или в плагинах. Scheduler привязывает незапущенные pod к нодам. Все остальные функции уровня кластера представлены в ControllerManager. Например, ноды обнаруживаются, управляются и контролируются средствами nodecontroller. Эта сущность в итоге может быть разделена на отдельные компоненты, чтобы сделать их независимо подключаемыми. ReplicationController – это механизм для запуска и поддержания заданного количества реплик pod в кластере.

Как упоминалось ранее, Kubernetes позволяет развертывать и управлять Docker контейнерами. Docker – программное обеспечение для автоматизации развёртывания и управления приложениями в среде виртуализации на уровне операционной системы. Docker позволяет решать следующие основные задачи:

- Запуск приложения изолированно друг от друга при помощи контейнеров;
- Отсутствие необходимости конфигурации среды для запуска (так как она поставляется вместе с приложением, упакованным в контейнер);
- Упрощает задачу масштабируемости приложения за счет системы оркестрации (Kubernetes).

Одним из вариантов запуска приложений на одном хосте для разных операционных систем является использование виртуальной машины. Однако такой подход обладает рядом недостатков, основным из которых является значительный расход ресурсов на поддержание функционирования гостевой операционной системы.

Альтернативным подходом является использование контейнеров [3, с. 867]. Здесь речь идет о виртуализации на уровне операционной системы. При этом используются такие Linux-механизмы как namespaces и controlgroups. Docker предоставил возможность удобного использования контейнеров на всех стадиях разработки программных средств.

Основными понятиями, о которых необходимо иметь представление при начале работы с Docker, являются образы и контейнеры [4, с. 826]. Как образы, так и контейнеры состоят из слоев. Слой – это набор файлов и директорий. К примеру можно рассмотреть как отдельные слои файлы ядра линукс, файлы какого-либо дистрибутива линукс, различные программные средства, файлы разрабатываемого приложения. Следует отметить, что если два образа используют один и тот же слой (например ядро линукс), то файлы данного слоя

загружаются лишь один раз и используются образами совместно. Основное отличие образа от контейнера заключается в том, что образ состоит из read-only слоев, в то время как контейнер – это запущенное приложение, которое в процессе может требовать прав записи на диск. Следовательно, в контейнере присутствует слой, который может использоваться как для чтения, так и для записи.

Чтобы создать образ, необходимо написать Dockerfile, который состоит из инструкций, добавляющих к образу новый слой. Создание образа на основе Dockerfile производится с помощью команды *dockerbuild*. Для запуска приложения на основе Dockerfile, необходимо запустить контейнер (т.е. предоставить приложению внутри образа слой для записи). Данное действие производится с помощью команды *dockerrun*. Просмотр всех образов в системе можно осуществить с помощью команды *dockerimages*. Посмотреть список контейнеров позволяет команда *dockercontainerls*.

Одной из основных возможностей Docker является возможность его запуска в режиме роя (swarmmode). Чтобы ввести Docker в данный режим, необходимо выполнить команду *dockerswarminit*. После этого необходимо запустить разрабатываемое приложение. Docker позволяет разбить его на несколько независимых частей (например http-сервер и сервер базы данных). Каждая часть распределенного приложения представляется сервисом (service). Сервис может использовать несколько реплик, что позволяет запускать столько экземпляров данной части приложения, сколько необходимо. Кроме этого, каждому сервису можно ограничить выделяемые ресурсы (память, процессорное время). Таким образом, сервис – это часть распределенного приложения, которая является аналогом контейнера при работе Docker в режиме роя. Масштабирование приложения производится путем изменения конфигурационных файлов. При этом нет необходимости останавливать разрабатываемое приложение, Docker берет на себя обязанности по развертыванию и запуску новой версии приложения.

Описанные выше технологии и фреймворки позволяют в значительной степени снизить трудозатраты на развертывание, мониторинг, обслуживание и управление кластером, в значительной степени упрощая выполнение этих процессов для разработчика кластерного веб-приложения. Применения данных решений позволяет повысить масштабируемость и надежность распределенных систем.

Библиографический список

1. Терских, М. Г. Технологии изоляции приложений и инструментальные средства для управления контейнерами / М. Г. Терских // Теория и практика современной науки. – 2017. – № 6 (24). – С. 825–829.
2. Сурай, О. В. Масштабирование микросервисной архитектуры на базе Kubernetes / О. В. Сурай // Актуальные научные исследования в современном мире. – 2018. – № 4–10 (36). – С. 118–122.
3. Афанасьев, Г. И. Создание кластера Postgresq в среде Kubernetes / Г. И. Афанасьев, М. М. Абулкасимов // Аллея науки. – 2018. – № 1 (17). – С. 840–844.
4. Орлов, З. А. Исследование средств масштабирования систем контейнерной виртуализации / З. А. Орлов // Аллея науки. – 2018. – № 9. – С. 867–871.

Г. Ю. Жулимов, А. Д. Мальцева, З. И. Баусова, К. М. Буданов

Пензенский государственный университет, Пенза, Россия

РЕШЕНИЕ ЭКОНОМИЧЕСКИХ ЗАДАЧ, СВЯЗАННЫХ С КОНФЛИКТНЫМИ СИТУАЦИЯМИ

Рассматривается теория игр, целью которой является поиск наиболее оптимального плана действия для каждого участника конфликтной ситуации.

В настоящее время часто приходится встречаться с ситуациями, требующими принятия решения в условиях неопределённости. Как правило, в таких задачах двое или более участников стремятся достичь различных целей, причём последствия всякого мероприятия

Нетрудно заметить, что приведённые задачи линейного программирования являются взаимно-двойственными, и при определении оптимальных стратегий в конкретных задачах предпочтительнее выбирать такую задачу, процесс решения которой будет наименее трудоёмким, и в таком случае решение более сложной задачи находится с помощью теорем двойственности.

Наряду с рассмотренным методом используется и следующий алгоритм решения произвольной конечной игры размера $m \times n$:

1. Устранить из матрицы игры те стратегии, которые по сравнению с другими не являются выгодными, то есть такие стратегии, с которыми соотносятся строки (столбцы) с элементами, явно меньшими (большими), чем соседние.

2. Найти нижнюю и верхнюю цены игры и узнать, имеет ли игра седловую точку [2]. Если седловая точка есть, то соответствующие ей стратегии игроков будут оптимальными, а цена совпадает с верхней (нижней) ценой.

3. Если седловая точка отсутствует, то решение следует искать в смешанных стратегиях. При этом задача решается симплексным методом, реализованном в MSExcel при помощи надстройки «Поиск решения».

Материалы данной статьи были использованы для разработки лабораторной работы на тему «Решение экономических задач с конфликтными ситуациями с использованием игровых моделей» по дисциплине «Прикладные методы оптимизации» для подготовки бакалавров по дисциплине 09.03.03 Прикладная информатика.

Библиографический список

1. Невежин В. П. Исследование операций и принятие решений в экономике. Сборник задач и упражнений : учеб. пособие для вузов / В. П. Невежин, С. И. Кружилов, Ю. В. Невежин ; под общ. ред. В. П. Невежина. – М. : ФОРУМ, 2012. – 400 с.

2. Косоруков, О. А. Исследование операций : учебник / О. А. Косоруков, А. В. Мищенко ; под общ. ред. д.э.н., проф. Н. П. Тихомирова. – М. : Экзамен, 2003. – 448 с.

3. Решение экономических задач с помощью математического моделирования: метод. указания к курсовой работе по дисциплине «Методы оптимизации» / сост.: З. И. Баусова, О. А. Голышевский, А. В. Еременко. – Пенза : Изд-во ПГУ, 2002. – 52 с.

Н. В. Сорокина, И. Ю. Балашова

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА АППАРАТНО-ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ЭЛЕКТРООБОГРЕВОМ ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ АЭС

Рассмотрена система управления электрообогревом технологического оборудования атомных электростанций, применяемая в случае использования в производственном цикле натрия в качестве теплоносителя. Описана архитектура аппаратного обеспечения данной системы. Разработан алгоритм управления исполнительным механизмом. Предложена программная реализация алгоритма.

В современном мире вопрос производства электроэнергии занимает важное место. Электрическая энергия используется практически во всех сферах нашей жизни. Невозобновляемость природных ресурсов, таких как нефть, газ, каменный уголь, заставляет задуматься о поиске альтернативных вариантов генерации электроэнергии. Помимо существующих тепловых и гидроэлектрических станций особое внимание заслуживают атомные электростанции.

Атомные электростанции (АЭС) представляют собой комплекс систем, устройств, оборудования и сооружений, предназначенный для производства электроэнергии. В производственном цикле тепло из активной зоны реактора отводится теплоносителем. При применении в качестве теплоносителя жидкого натрия важное место занимает система управления электрообогревом технологического оборудования АЭС. Электрообогрев предназначен для разогрева оборудования и трубопроводов установки перед заполнением рабочей средой расплавом натрия, поддержания их в разогретом состоянии при эксплуатации, разогрева перед дренированием рабочей среды, разогрева после замораживания рабочей среды, а также для обеспечения режима регенерации и снижения потерь тепла с оборудования и трубопроводов. Электрообогрев осуществляется посредством нагревателей, объединяемых в тепловые зоны.

Ни один сложный технический объект не может обойтись без автоматизации управления на базе компьютерной техники. Необходимость использования автоматизированных систем управления (АСУ) и систем автоматического управления (САУ) на базе микропроцессоров и микроконтроллеров обусловлена тем, что человек не может физически без их помощи справиться с задачами управления, заключающимися в обработке большого объема информации и формирования на ее основе управляющих воздействий на объект в масштабе реального времени. Ядром САУ и АСУ является их программное обеспечение (ПО), которое имеет практическую ценность и реализует необходимую функциональность.

Система управления электрообогревом технологического оборудования атомных электростанций является одной из разновидностей систем программно-технического комплекса (ПТК) «УРАЛ-Атом», предназначенного для создания и модернизации систем контроля и управления технологическим оборудованием. Архитектура ПТК «УРАЛ-Атом» представлена на рис. 1.

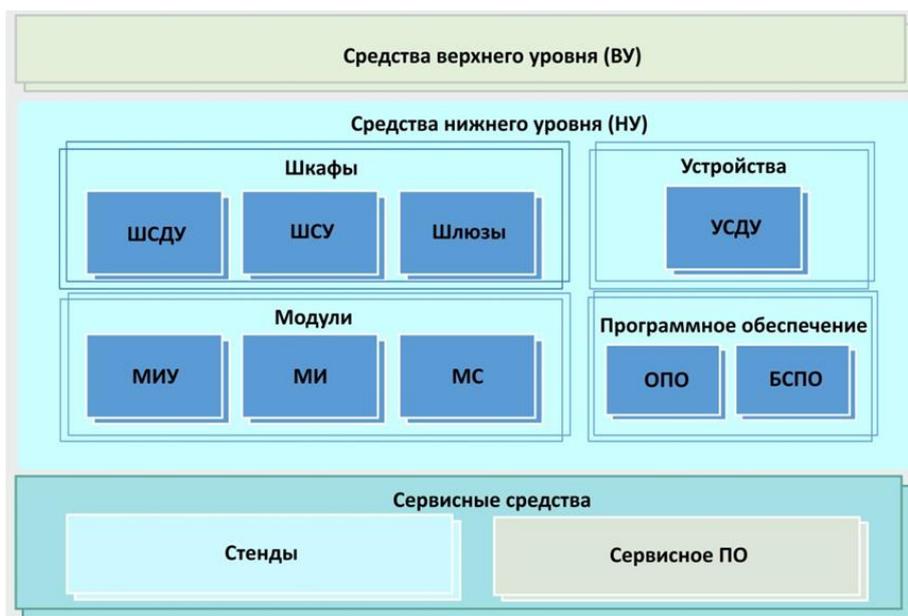


Рис. 1. Архитектура ПТК «УРАЛ-Атом»

В состав компонентов верхнего уровня (ВУ) входят рабочие станции и серверы, объединенные в локальную вычислительную сеть верхнего уровня. Подсистема ВУ обеспечивает контроль и управление технологическим объектом со стороны эксплуатирующего персонала.

В состав компонентов нижнего уровня (НУ) входят:

- шкафы сбора данных и управления (ШСДУ);
- шкафы силового управления (ШСУ);
- шлюзы;

- устройства сбора данных и управления (УСДУ);
- модуль индивидуального управления (МИУ);
- модуль информационный (МИ);
- модуль серверный;
- общее программное обеспечение (ОПО);
- базовое специальное программное обеспечение (БСПО).

В задачи подсистемы нижнего уровня входит: сбор данных о технологическом объекте управления от датчиков, обработка данных, управление исполнительным механизмом в автоматизированном режиме.

В состав программного обеспечения входит общее программное обеспечение (ОПО) и базовое специальное программное обеспечение (БСПО). ОПО создано на основе операционной системы с открытыми исходными кодами. БСПО разработано на основе методологии объектно-ориентированного программирования в виде библиотеки классов.

Программно-технический комплекс оснащен вспомогательными средствами, такими как технологические стенды и сервисное программное обеспечение.

Модули индивидуального управления обеспечивают управление технологическим оборудованием в соответствии с установленными алгоритмами на основе данных о состоянии.

Модули информационные предназначены для сбора аналоговых и дискретных сигналов от датчиков состояния технологического объекта управления и передачи собранной информации другим модулям системы по цифровым каналам связи.

Модули серверные обеспечивают взаимодействие модулей индивидуального управления и модулей информационных со шлюзом, стендами и другими устройствами системы. Сопряжение обеспечивается по сетевым интерфейсам RS-485 и Ethernet, используются транспортные протоколы Modbus, TCP, IP, UDP.

Центральное звено модуля автоматизации – микроконтроллер, организующий функционирование модуля. В микроконтроллер загружается программное обеспечение, реализующее необходимую функциональность.

В разработанном программном обеспечении одним из базовых объектов является объект класса подзадача. При создании этого объекта запускаются функции, выполняющиеся как процессы операционной системы. Передача управления между ними осуществляется в режиме разделения времени. Количество и тип таких концептуальных процессов (потоков) зависит от количества и типов подзадач, размещаемых на данном модуле автоматизации (в рамках одного физического процесса). В ходе разработки программного обеспечения были выделены следующие виды подзадач:

- подзадача управления исполнительным механизмом (ИМ);
- подзадача информационного взаимодействия;
- подзадача функционального контроля и диагностики ИМ.

Целью подзадачи управления исполнительным механизмом является автоматизированное управление исполнительным механизмом в соответствии с заданным алгоритмом.

Целью подзадачи информационного взаимодействия является сбор и хранение технологических параметров объекта и среды управления, их предварительная обработка и обеспечение данными о текущем состоянии другие подзадачи.

Целью подзадачи функционального контроля и диагностики является непрерывное диагностирование состояния аппаратных и программных средств и выдача соответствующей информации на верхний уровень.

Одной из важнейших является подзадача управления исполнительным механизмом. Созданный рабочий цикл выполнения подзадачи управления исполнительным механизмом можно представить в виде последовательности команд. Сначала выполняется запрос текущих значений технологических переменных (температуры) и состояния исполнительного механизма (нагревателя). В соответствии со шкалой приоритетов видов управления первым вызывается объект технологическая защита, которому передается текущий набор входных значений. Объект выполняет анализ. Если условия срабатывания технологической защиты выполняются (обрыв цепи нагревателя или повреждение изоляции), то выдается команда

«выключить» объекту класса исполнительный механизм – нагревателю, устанавливается признак «технологическая защита», выдается донесение на верхний уровень и выполнение текущей итерации цикла завершается. Если условия не выполняются, то вызывается объект блокировки, которому передается текущий набор входных значений. После анализа поступивших к объекту данных, возможно два варианта развития событий. Если выполняются условия блокировки (блок в тестовом положении или выключатель автоматический отключен), то запрещаются все воздействия на объект класса исполнительный механизм, в классе устанавливается признак «блокировка», выдается донесение на верхний уровень и выполнение текущей итерации цикла завершается. Если условия не выполняются, то проверяется факт выдачи команды дистанционного управления с верхнего уровня на исполнительный механизм. Если признак установлен, то выдается донесение на верхний уровень и выполнение текущей итерации цикла управления завершается. Иначе проверяется факт выдачи команды функционально-группового управления на исполнительный механизм. Если признак установлен, то выдается донесение на верхний уровень и выполнение текущей итерации цикла управления завершается, в противном случае – вызывается объект автоматическое регулирование. После анализа объектом поступивших данных, возможно два варианта. Если в соответствии с алгоритмом и уставками выполняются условия автоматического регулирования, то выдается (не выдается) команда (включить или выключить) объекту класса исполнительный механизм, выдается донесение на верхний уровень и выполнение текущей итерации цикла завершается. В случае недостоверных параметров, превышения допустимых значений автоматическое управление приостанавливает работу с установлением соответствующего признака в объекте класса исполнительный механизм и выдаче донесения о необходимости перехода в режим дистанционного управления. Выполнение текущей итерации цикла управления завершается. Цикл работает бесконечно с периодом 100 мс. В случае недостоверных параметров будет работать технологическая сигнализация или блокировка, в соответствии с алгоритмом, и выдаваться донесения о текущем состоянии до устранения этих причин.

Аппаратно-программное обеспечение предназначено для автоматизированного управления нагревателем из состава системы электрообогрева технологического оборудования в соответствии с заданным алгоритмом.

Библиографический список

1. Давиденко, К. Я. Технология программирования АСУ ТП / К. Я. Давиденко. – М. : Энергоатомиздат, 1986. – С. 184.
2. Мякишев, Д. В. Принципы и методы создания надежного программного обеспечения АСУТП / Д. В. Мякишев. – М. : Инфа-Инженерия, 2017. – С. 114.
3. АЭС // РОСАТОМ. – URL: <http://www.rosatom.ru/about-nuclear-industry/powerplant/> (дата обращения: 10.09.2018).

И. Ю. Балашова

Пензенский государственный университет, Пенза, Россия

ИНТЕЛЛЕКТУАЛИЗАЦИЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО УПРАВЛЕНИЮ ПЕРСОНАЛОМ ОРГАНИЗАЦИИ НА ОСНОВЕ НЕЧЕТКО-ОНТОЛОГИЧЕСКОЙ МОДЕЛИ

Построена модель поддержки принятия решений по подбору персонала, позволяющая учесть семантически неопределенные формулировки требований к соискателю. Формализованы нечетко сформулированные требования к соискателю. Описано модельное представление рассуждений менеджера по персоналу, учитывающее семантику неопределенности.

В управлении персоналом одним из наиболее ответственных этапов является этап подбора кадров. Качество подобранных кандидатов на вакантные должности напрямую влияет на результат работы организации. На российском рынке информационных технологий присутствует достаточно большой выбор программных продуктов, позволяющих автоматизировать отдельные этапы подбора кадров, например, «1С: Зарплата и Управление Персоналом 8.0», E-Staff Рекрутер, PersonelManager, «БОСС-Кадровик» и др. Однако существует ряд причин, по которым эффективность существующих систем не востребована в полной мере. К главной причине можно отнести невозможность известными на рынке системами давать оценку соответствия потенциальным кадрам при семантически нечетко формулируемых требованиях к вакансии. Использование гибридного нечетко-онтологического и логического вывода интеллектуальной информационной системе позволит решить задачу управленческого решения в условиях неполной и нечеткой входной информации, наиболее полно отразить особенности естественного языка при диалоговом взаимодействии.

Разработана нечеткая предметная OWL-онтология

$$\tilde{O} = \langle I, C, P, A, L, F, Mod \rangle,$$

где I – множество экземпляров классов онтологии; C – множество нечетких классов онтологии; P – множество нечетких свойств объектов; A – множество нечетких аксиом онтологии; L – множество лингвистических переменных; F – множество операторов нечетких логик соответствующих типов; Mod – множество нечетких модификаторов, т.е. функций модификации функций принадлежности, нечетких классов и нечетких отношений.

Обозначим $T = \{T_1, T_2, \dots, T_n\}$, $T \subset C$ – подмножество классов онтологии, описывающих требования, предъявляемые к кандидату на должность. В множестве T выделим требования, имеющие нечеткий характер: возраст соискателя, стаж работы, уровень владения необходимыми компетенциями в профессиональной области и др. Данным требованиям в рассматриваемой модели соответствуют лингвистические переменные

$$\langle T_i, M_i, X_i, G, P_i \rangle,$$

где T_i – наименование лингвистической переменной; M_i – терм-множество лингвистической переменной T_i , т.е. множество ее значений (термов), являющихся нечеткими переменными \tilde{T}_i^j , $j = \overline{1, k_i}$; X_i – область определения нечетких переменных, входящих в определение лингвистической переменной T_i ; G – процедура образования новых термов с помощью логических связок «И», «ИЛИ» и модификаторов типа «ОЧЕНЬ», «НЕ» и др.; P_i – процедура задания на множестве X_i соответствующих нечетких переменных.

Требуется установить рейтинг соискателя, т.е. степень соответствия характеристик соискателя предъявляемым требованиям. Непосредственный процесс интеграции онтологической и продукционной моделей знаний предполагает использование в качестве объектов продукционных правил аналогичные объекты классов, описанных в построенной нечеткой OWL-онтологии.

Рейтинг соискателя представляет собой выходную лингвистическую переменную R и описывается подобным образом. Тогда система нечеткого вывода будет содержать конечное множество правил нечетких продукций вида:

ПРАВИЛО $\langle k \rangle$:

$$\text{ЕСЛИ } \langle T_1 \text{ есть } \nabla \tilde{T}_1^j \rangle \Theta \dots \Theta \langle T_n \text{ есть } \nabla \tilde{T}_n^p \rangle \text{ ТО } \langle R \text{ есть } \nabla \tilde{R}_q \rangle (\alpha_k),$$

где Θ – некоторая из бинарных операций нечеткой конъюнкции «И» или нечеткой дизъюнкции «ИЛИ»; ∇ – модификатор, определяемый процедурами G и P_i ; $\alpha_k \in [0; 1]$ – весовой коэффициент, устанавливающий степень значимости каждого правила.

При разработке схемы нечеткого вывода использован алгоритм Мамдани, который в настоящее время получил наибольшее практическое применение. Численным значением входных лингвистических переменных является численное значение соответствующей ха-

рактеристики соискателя. Обозначим t_i – известное количественное значение i -й входной лингвистической переменной, t'_i – результат фаззификации соответствующего подусловия правила из базы правил. Тогда результат фаззификации i -го подусловия

$$t'_i = \mu(t_i),$$

где $\mu(t_i)$ – значение функции принадлежности термина соответствующей входной лингвистической переменной.

Для нахождения степени истинности условий каждого из правил нечетких продукций применяются парные логические операции. Результат нечеткой конъюнкции определяется по формуле алгебраического произведения, нечеткой дизъюнкции – по формуле алгебраической суммы, при этом значения t'_i используются в качестве аргументов соответствующих логических операций. В результате находятся все значения t''_k для каждого k -го правила, входящего в базу правил. Те правила, степень истинности которых отлична от нуля, считаются активными и используются для дальнейших расчетов.

В базе нечетких правил заключения заданы в форме элементарных нечетких лингвистических высказываний, поэтому степень истинности заключения r_k каждого из активных правил соответствующей базы будет совпадать со степенью истинности соответствующего условия t''_k : $r_k = t''_k$. После нахождения множества степеней истинности каждого заключения базы правил определяются функции принадлежности каждого из заключений для выходной лингвистической переменной с помощью min-активизации:

$$\mu'_k(r) = \min\{r_k, \mu_k(r)\},$$

где $\mu_k(r)$ – функция принадлежности термина выходной лингвистической переменной в k -м правиле; $\mu'_k(r)$ – функция принадлежности активизированного термина выходной лингвистической переменной в k -м правиле. Таким образом, для выходной лингвистической переменной определяются функции принадлежности нечетких множеств ее значений в каждом из активных правил. Результат аккумуляции для выходной лингвистической переменной определяется как объединение нечетких множеств по формуле:

$$\mu'(r) = \max_k\{\mu'_k(r)\},$$

где $\mu'(r)$ – итоговая функция принадлежности выходной лингвистической переменной. Результатом дефаззификации выходной лингвистической переменной является количественное значение рейтинга соответствия соискателя вакансии r' , получаемое на основе метода центра тяжести:

$$r' = \frac{\int_0^{100} r \cdot \mu'(r) dr}{\int_0^{100} \mu'(r) dr},$$

Полученное значение рейтинга используется при формировании списка наиболее подходящих претендентов. Сортировка списка по рейтингу позволит установить лучших кандидатов на вакантную должность.

Взаимосвязь самой онтологии предметной области и набора правил в процессе логического вывода осуществляется с помощью формирования запросов к онтологии, генерируемых системой анализа при выполнении набора правил. Процесс вывода представляет собой движение резонера по графу, в узлах которого находятся объекты соответствующих классов онтологий, удовлетворяющих или не удовлетворяющих заданным условиям.

Таким образом, применение онтологического и нечеткого анализа при разработке интеллектуальной системы управления персоналом позволяет осуществлять более качественный отбор кандидатов на вакансии благодаря возможности согласовать формальные критерии и неформальные требования к области потенциальных кандидатов или вакансий.

Библиографический список

1. Вязигин, А. В. Оценка персонала высшего и среднего звена / А. В. Вязигин. – М. : Вершина, 2006. – 256 с.
2. Баскина, Т. В. Техники успешного рекрутмента / Т. В. Баскина. – М. : Альпина Паблишер, 2014. – 288 с.
3. Белов, В. Н. Выбор функции обработки данных для оценки деятельности сотрудников вуза / В. Н. Белов, П. П. Макарычев // Информационные системы и технологии. – 2012. – № 1 (69). – С. 45–50.

Е. С. Ефимова, Я. А. Колесникова

Пензенский государственный университет, Пенза, Россия

**САМЫЕ РАСПРОСТРАНЕННЫЕ ERP-СИСТЕМЫ
В РОССИЙСКОЙ ФЕДЕРАЦИИ**

Приводится описание ERP-систем. Составлен список лидеров российского рынка ERP-систем, которые показали неплохие результаты за годы своего существования. Актуальность данной статьи не вызывает сомнения, поскольку в настоящее время все большее внимание уделяется автоматизации информационных процессов на предприятии.

Информационные системы управления все плотнее входят в деятельность современных предприятий. Их разновидностей очень много, и ежегодно можно услышать о новых продуктах ERP.

Аббревиатура ERP происходит от английского выражения «Enterprise Resource Planning», что, в свою очередь, означает планирование ресурсов предприятия. Система представляет собой общую стратегию деятельности компании, в основе которой лежит принцип единого хранилища данных, содержащий всю корпоративную бизнес-информацию и учитывающий такие направления, как управление трудовыми ресурсами и активами компании, управление финансовыми ресурсами, взаимодействие с партнерами и клиентами. Данное хранилище устраняет необходимость передачи данных от одной системы к другой, так как вся взаимосвязанная информация находится в одной базе данных, и обеспечивает единовременный доступ к информации любого числа сотрудников предприятия.

Внедрение ERP системы на предприятия достаточно долгий и затратный процесс, поэтому важно отнестись со всей ответственностью к выбору программного продукта. Сделать правильный выбор позволяет анализ критериев ERP-систем, самые распространенные критерии связаны со стоимостью системы, учитывая соотношение «цена – качество», «цена – функционал». Кроме этого, высоко оцениваются интегрируемость, гибкость и масштабируемость системы наряду с открытостью, возможностью расширения имеющегося функционала путем дополнительных разработок и адаптации системы под специфические нужды предприятия.

Рассмотрим самые распространенные и проверенные системы ERP в Российской Федерации:

– SAP AG. Система R/3, разработанная крупнейшим немецким производителем программного обеспечения SAP AG, известна на российском рынке с 2003 года. Компания постоянно осуществляет комплекс мероприятий, направленных на продвижение продукта, обучение консультантов и приобретение опыта в различных отраслях экономики. Самым ярким пользователем немецкой системы выступает российский «Лукойл».

В основе данной программы лежит «Базисная система». Она представляет собой большой электронный склад информации, к которому подключены прикладные модули, что позволяет не зависеть от аппаратной платформы. Система функционирует под такими ОС, как Windows, Unix, OS/2, Macintosh. SAP R3 заслуженно считается одной из самых

масштабных систем ERP. Стоимость поставки системы на 50 рабочих мест составляет \$350. Стоимость внедрения – в несколько раз выше стоимости лицензии.

– Oracle. Эта система является прямым конкурентом SAP R3. Она представляет собой полный комплекс различных приложений для автоматизации бизнеса. С помощью этой системы можно решать бизнес-задачи от самых простых до самых сложных. В нее входит 150 модулей, рассчитанных на разные сферы бизнес-деятельности. Их можно распределить на три основных блока:

- Oracle ERP – производственный процесс, контроль финансов, снабжение.
- Oracle CRM – маркетинг, продажи и т.д.
- Oracle E-Hub – электронные торговые площадки.

У данного продукта есть единое хранилище информации, как и у вышеописанной SAP R3.

– MicrosoftBusinessSolutions. Компания Microsoft является одной из ведущих поставщиков ERP-систем в мире, официально она представлена в России с ноября 1992 г.

Предлагаемая Microsoft линейка продуктов для комплексной автоматизации ориентирована в первую очередь на предприятия малого и среднего бизнеса и включает такие системы, как Ахарта, GreatPlains, Navision, Solomon, Microsoft CRM. На базе Microsoft Ахарта был создан ряд отраслевых ERP-систем, в частности отраслевое решение АхартаRetail, адаптированное к особенностям российских розничных сетей. Стоимость поставки и внедрения системы на 1 рабочее место составляет €1600-2500. Пакет на 20 пользователей стоит €36 000–50 000.

– Корпорация Галактика. Галактика ERP полнофункциональный комплекс бизнес-решений Галактика BusinessSuite, построенный на ядре, был предложен вниманию клиентов в 2005 г.

Эта система – продукт российских разработчиков. Это является ее большим преимуществом, так как при ее создании были учтены все особенности российской бизнес-деятельности. Кроме того, «Галактика» – одна из самых распространенных систем, используемая на предприятиях в России. Особой популярностью она пользуется в топливной, химической, нефтехимической и пищевой отраслях.

– 1С:ERP Управление предприятием. Со дня своего основания в 1991 г. компания «1С» разработала целый ряд бизнес-решений, получивших широкое распространение на компьютерном рынке. 1С позволит построить комплексную информационную систему для управления деятельностью любого предприятия. Это инновационное решение от компании «1С» использует комплексный подход к управлению бизнесом, лучшие международные методики и многолетнюю отечественную практику, что гарантирует гибкость настройки, удобство использования и существенный экономический эффект. Линейка решений «1С:ERP» охватывает все основные отрасли, имеет большой набор функций и программных инструментов, подходит для использования на предприятиях любой численности. Более 885 000 рабочих мест автоматизировано на ERP-решениях «1С», а общая численность персонала клиентов превышает 14 миллионов человек.

Подводя итог, стоит отметить, что внедрение ERP-системы – удовольствие затратное как по времени, так и по финансам. Но если верить статистике, то можно понять, что при правильном подходе и с командой профессионалов внедрение не только окупится в течение первого года эксплуатации, но и в дальнейшем начнет приносить свои плоды.

По мнению большинства экспертов, количество внедрений является наиболее объективным критерием оценки спроса на ERP. По данному показателю вне конкуренции остаётся отечественная система «1С:ERP Управление предприятием». За 2017 год было зарегистрировано 167 новых проектов внедрения данного программного продукта, а рыночная доля ERP-систем на платформе «1С:Предприятие» достигает 31 %. Немаловажным фактором является то, что внедрение «1С:ERP» можно начать с необходимых модулей и со временем подключать остальные. Огромным плюсом для некоторых компаний становится относительно невысокая стоимость продукта, лицензия на внедрение типового функционала составляет 400 000 руб.

Библиографический список

1. Что такое ERP-система? Планирование финансовых ресурсов предприятия. Полезные советы и интересные факты. – URL: <http://fb.ru/article/155116/chto-takoe-erp-sistema-planirovanie-finansovyih-resursov-predpriyatiya>
2. Проблема выбора ERP-системы. Научная электронная библиотека «КиберЛенинка». – URL: <https://cyberleninka.ru/article/v/problema-vybora-erp-sistemy>
3. Обзор российского рынка ERP 2017. СофтЭксперт. – URL: <http://www.sfx-tula.ru/news/infoblog/8971/>
4. Что же такое ERP-система? Бесплатное рекламное пространство. – URL: http://www.catalogy.ru/node10_biznes_4525/chto-zhe-takoe-erp-sistema.htm
5. ERP-системы. 1С Предприятие. ERP-портал. – URL: <http://www.erp-online.ru/software/1c/>

Л. Р. Фионова, М. А. Катыхева

Пензенский государственный университет, Пенза, Россия

**ДОКУМЕНТИРОВАНИЕ ЭЛЕКТРОННЫХ ТОРГОВ
ДЛЯ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ**

Рассматривается процедура проведения электронных торгов для медицинских учреждений. Перечислены и описаны виды документов для электронных торгов, обоснована роль информационных технологий в осуществлении закупочной деятельности.

В связи с принятием и реализацией государственной программы «Цифровая экономика РФ» (распоряжение Правительства РФ от 28.07.2017 № 1632-р) информационные технологии должны внедряться во все направления деятельности различных организаций и учреждений, в том числе и медицинских.

В целях повышения эффективности размещения государственного заказа, а также для противодействия коррупции в сфере государственного финансирования еще в 2005 году Правительством Российской Федерации был принят ряд федеральных законов [1, 2]. С 1 января 2011 года заказчики субъектов Российской Федерации и муниципальные заказчики стали обязаны проводить открытые аукционы в электронной форме.

Основным законом, который регулирует проведение электронных торгов для медицинских учреждений в настоящее время, является Федеральный закон «О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд» от 05.04.2013 № 44-ФЗ (далее – ФЗ № 44) [3].

Рассмотрим документирование электронных торгов на примере Сердобской межрайонной больницы (далее – больницы). Для осуществления закупок в больнице в соответствии со статьей 38 ФЗ № 44 в больнице создана контрактная служба. В состав работников контрактной службы входят в основном сотрудники отдела материального технического снабжения.

Закупочная деятельность любой организации, в том числе и больницы, включает следующие этапы: планирование и обоснование закупок: составление, утверждение и исполнение плана закупок и планов-графиков; осуществление закупок: определение и отбор поставщиков, подготовка закупочной документации; заключение контрактов и их исполнение; контроль закупок.

Стратегическим плановым документом с целью определения направлений расходования средств на закупки товаров (работ, услуг) в больнице является план закупок. Он формируется на текущий финансовый год и на плановые последующие два года. При подготовке данного документа сотрудники контрактной службы больницы используют форму плана закупок, регламентированную Постановлением Правительства РФ [4].

Обоснование закупок отражается также в плане-графике, который создается ежегодно на основании плана закупок. Предварительно для составления проекта плана-графика в отдел материально-технического снабжения из отделений больницы поступают служебные записки с заявками по закупкам не позднее 15 декабря текущего года. При формировании плана-графика работники контрактной службы больницы руководствуются статьей 18 ФЗ № 44 и обосновывают начальную (максимальную) цену контракта (НМЦК), цену контракта, а также выбранный способ определения поставщика (подрядчика, исполнителя) и, в том числе, дополнительные требования к участникам закупки. Его форма также регламентирована Постановлением Правительства РФ [5].

Таким образом, план-график является обязательным документом и отражает последовательную детализацию плана закупок, а именно: описание закупки; дату поставки; дату оплаты; источник финансирования. Работник контрактной службы публикует план-график в течение трех рабочих дней с момента его утверждения в единой информационной системе (ЕИС) в сфере закупок на сайте www.zakupki.gov.ru. Эта система служит хранилищем всей информации, которая касается закупок товаров или услуг для нужд государственных и муниципальных организаций.

В течение всего года работа по определению поставщиков и заключению с ними контрактов осуществляется по утвержденному плану-графику.

Далее работник контрактной службы в соответствии со статьей 63 ФЗ № 44 подготавливает извещение о проведении электронного аукциона вместе с документацией и публикует его в ЕИС. В зависимости от суммы НМЦК определяется срок аукциона. Если сумма НМЦК превышает 3 млн руб., то извещение размещается в ЕИС за 15 дней до дня окончания приема заявок, а если сумма меньше 3 млн руб., то за 7 дней. В случае, если принимается решение о внесении изменений в заказ, то работник контрактной службы больницы в течение двух рабочих дней после размещения извещения готовит проект письма о внесении изменений. После подписания письма он в течение одного рабочего дня публикует изменения на электронной площадке в ЕИС.

В извещении о проведении электронного аукциона указывается следующая информация:

- адрес электронной площадки в информационно-телеком-муникационной сети Интернет;
- дата окончания срока рассмотрения заявок на участие в аукционе;
- дата проведения такого аукциона. День проведения аукциона переносится на следующий за ним рабочий день, если дата проведения аукциона приходится на нерабочий день;
- размер обеспечения заявок на участие в аукционе;
- преимущества, которые предоставляет заказчик;
- требования, предъявляемые к участникам аукциона, и исчерпывающий перечень документов, которые представляют участники аукциона [3].

К извещению работник контрактной службы готовит документацию об электронном аукционе, в которой указывает следующие сведения:

- наименование и описание объекта закупки и условия контракта, в том числе обоснование НМЦК;
- требования к содержанию, составу заявки на участие в аукционе и инструкция по ее заполнению. Требования не устанавливаются, если их исполнение ограничивает количество участников аукциона или ограничивает доступ их к участию в аукционе;
- дата и время окончания срока подачи заявок на участие в аукционе;
- дата окончания срока рассмотрения заявок на участие в аукционе;
- дата проведения аукциона;
- сведения о валюте, которая используется для определения цены контракта и расчетов с поставщиками;
- порядок применения официального курса иностранной валюты к рублю Российской Федерации, которая установлена Центральным банком Российской Федерации и используется при оплате контракта;

- размер обеспечения исполнения контракта, срок и порядок предоставления указанного обеспечения, требования к обеспечению исполнения контракта;
- возможность заказчика изменить условия контракта;
- информация о контрактной службе, срок, в течение которого с выигравшим аукцион заказчиком заключается контракт при уклонении им от заключения контракта, подписывается контракт, условия признания победителя аукциона или участника аукциона, который уклонился от заключения контракта;
- порядок, даты начала и окончания срока предоставления участникам аукциона разъяснений положений документации об аукционе;
- информация о возможности одностороннего отказа от исполнения контракта [3].

Документация об электронном аукционе также в обязательном порядке содержит проект контракта.

После размещения документации об электронном аукционе на площадке для формирования участников аукциона с целью определения НМЦК работник контрактной службы больницы подготавливает письма-запросы о ценовой информации пяти поставщикам. По окончании срока подачи заявок проводится аукцион в течение 10-15 дней. Из поступивших коммерческих предложений работник контрактной службы отбирает три из них, которые соответствуют предъявленным требованиям и в которых указана наименьшая цена поставки. На основе выбранных коммерческих предложений он формирует НМЦК, используя метод сопоставимых рыночных цен (анализ рынка), что соответствует требованиям ч. 1–6 ст. 22 ФЗ № 44 [3].

Таким образом, после выполнения всех этапов конкурсная документация на осуществление закупки больницы включает следующие документы: заявка на размещение; приложение № 1 к заявке (описание товара, ед. измерения); приложение № 2 к заявке (коды Общероссийского классификатора продукции по видам экономической деятельности, количество товара); письмо о составе аукционной комиссии; проект контракта; обоснование НМЦК; три коммерческих предложения; пять запросов для поставщиков; приложение к запросам (техническое задание с описанием товара, количества, единиц измерения).

Все документы брошюруются и к этой документации оформляются два сопроводительных письма и титульный лист. Утвержденная главным врачом больницы и согласованная с Министерством здравоохранения Пензенской области и Министерством финансов Пензенской области конкурсная документация на размещение государственного заказа публикуется вместе с извещением в единой информационной системе (ЕИС) в сфере закупок на сайте www.zakupki.gov.ru.

Результаты рассмотрения заявок на участие в электронном аукционе отражаются в протоколе. В указанной в извещении дате участники, допущенные к аукциону, начинают торги. Выигрывает тот участник, который предложил наиболее низкую цену и соответствующую всем требованиям документации. Результаты подведения аукциона отражаются в протоколе подведения итогов электронного аукциона. После публикации протокола итогов аукциона в ЕИС работник контрактной службы больницы готовит проект контракта с поставщиком. Процедура заключения контракта занимает не более 20 рабочих дней. Подписанные контракты передаются в бухгалтерию больницы. Как показывает практика, процесс проведения электронного аукциона максимально может занимать 37 дней. Работники контрактной службы больницы все заключенные контракты заносят в реестр закупок, который ведется в электронном виде и содержит следующую информацию: наименование объекта закупки, название и адрес местонахождения поставщика, дата закупки, цена закупки.

Как видно, выполнение каждого этапа закупочной деятельности сопряжено не только со значительными трудовыми и временными затратами, но и оформлением большого количества конкурсной документации, что оправдывает применение информационных технологий.

Таким образом, развитие цифровой экономики затронуло и медицинскую сферу, позволяет повысить эффективность закупочной деятельности. А от сотрудников контрактной службы больницы требуются владение информационными технологиями, умения пользоваться электронной подписью, находить информацию в Интернете, размещать заявки и загружать документы в единую информационную систему.

Библиографический список

1. О размещении заказов на поставки товаров, выполнение работ, оказание услуг для государственных и муниципальных нужд : федер. закон № 94-ФЗ от 21.07.2005.
2. О пользовании сайтами в сети Интернет, на которых осуществляется проведение открытых аукционов в электронной форме : постановление Правительства РФ № 179 от 17.03.2008.
3. О контрактной системе в сфере закупок товаров, работ, услуг для обеспечения государственных и муниципальных нужд : федер. закон № 44-ФЗ от 05.04.2013 (с изм. от 29.07.2017 № 231-ФЗ).
4. Об утверждении Правил формирования, утверждения и ведения плана закупок товаров, работ, услуг для обеспечения федеральных нужд, а также требований к форме плана закупок товаров, работ, услуг для обеспечения федеральных нужд : постановление Правительства РФ № 552 от 05.06.2015 (с изм. от 25.01.2017 № 73).
5. Об утверждении Правил формирования, утверждения и ведения плана-графика закупок товаров, работ, услуг для обеспечения федеральных нужд, а также требований к форме плана-графика закупок товаров, работ, услуг для обеспечения федеральных нужд : постановление Правительства РФ № 553 от 05.06.2015.

Л. Р. Фионова

Пензенский государственный университет, Пенза, Россия

УПРАВЛЕНИЕ ЭЛЕКТРОННЫМ ДОКУМЕНТООБОРОТОМ В ОРГАНАХ ВЛАСТИ

Рассматриваются нормативные акты, регламентирующие электронный документооборот в органах власти. Описаны правила организации электронного документооборота, сформулированы существующие проблемы.

Руководитель каждого органа власти должен владеть информацией о работе его подчиненных и быстро формулировать решения по устранению критических ситуаций и организации взаимодействия сотрудников. Информация может поступать на различных носителях. По типу носителя все документы можно разделить на бумажные, электронные и документы, использующие иные носители. Электронные документы в свою очередь могут быть текстовыми (со структурированным и неструктурированным текстом), графическими и мультимедийными. Быстрому предоставлению информации способствует внедрение и использование электронного документооборота (ЭД).

Можно выделить особенности современного состояния документационного обеспечения управления (ДОУ) в России (которые нужно учитывать при внедрении ЭД в конкретном органе власти):

- 1) одновременное существование бумажного и электронного документа и документооборота;
- 2) рекомендательный характер стандартов в России в соответствии с Федеральным законом «О техническом регулировании» от 27.12.2002 № 184-ФЗ;
- 3) необходимость соответствия каждого вновь издающегося национального стандарта соответствующему международному стандарту;
- 4) реализация Федеральных целевых программ «Электронная Россия», «Электронное Правительство» и «Информационное общество», «Цифровая экономика России»;
- 5) введение единых «Правил делопроизводства в федеральных органах исполнительной власти», утвержденных впервые в «перестроечной» России Постановлением Правительства РФ от 15.06.2009 № 477;
- 6) вывод Федерального архивного агентства из структуры Министерства культуры Российской Федерации.

В Концепции формирования в Российской Федерации электронного правительства [1] зафиксировано:

«К приоритетным направлениям государственной политики в области использования информационно-коммуникационных технологий отнесено:

- создание единой инфраструктуры обеспечения юридически значимого электронного взаимодействия;
- развитие защищенной межведомственной системы ЭД;
- формирование необходимой нормативной правовой базы формирования электронного правительства».

После утверждения вышеназванных Правил делопроизводства для дальнейшего формирования нормативной базы ЭД принято постановление Правительства РФ от 22.09.2009 № 754 «Об утверждении Положения о системе межведомственного ЭД (МЭДО)».

Положение о системе МЭДО является нормативной основой для создания единой инфраструктуры обеспечения юридически значимого электронного взаимодействия органов государственной власти и органов местного самоуправления.

Система МЭДО представляет собой защищенную технологическую систему, обеспечивающую взаимодействие федеральных информационных систем ЭД.

На первом этапе участниками МЭДО являются федеральные органы государственной власти, Администрация Президента России и Аппарат Правительства Российской Федерации.

МЭДО позволяет участникам:

- направлять и получать в электронной форме решения и поручения Президента и Правительства России;
- получать информацию о ходе рассмотрения федеральными органами исполнительной власти поручений Президента и Правительства России;
- направлять в электронной форме доклады Президенту и Правительству России;
- вносить в электронной форме в Правительство России проекты нормативных правовых актов, подготовленные федеральными органами исполнительной власти;
- проводить в электронной форме согласительные процедуры по проектам нормативных правовых актов.

Система МЭДО позволяет осуществлять информационный обмен посредством электронных сообщений, в состав которых на первом этапе включаются электронные образы документов, первоначально созданных на бумажном носителе.

В дальнейшем предполагается переходить на обмен документами, не имеющими аналога (подлинника) на бумажном носителе и создаваемыми исключительно в электронной форме. Приказом Росархива утверждены рекомендации, регламентирующие такой переход [2].

Для регулирования отношения в области использования электронных подписей при оказании государственных и муниципальных услуг, исполнении государственных и муниципальных функций, при совершении иных юридически значимых действий необходимо соблюдать требования Федерального закона от 06.04.2011 № 63-ФЗ. В этом законе зафиксировано, что «виды электронных подписей, используемых органами власти, порядок их использования, а также требования об обеспечении совместимости средств электронных подписей при организации электронного взаимодействия указанных органов между собой устанавливает Правительство России».

Статья данного закона фиксирует условия признания электронных документов, подписанных электронной подписью, равнозначными документам на бумажном носителе, подписанным собственноручной подписью. Эти условия необходимо учитывать при организации ЭД.

Электронный документ должен быть оформлен по общим правилам делопроизводства и иметь реквизиты, установленные для аналогичного документа на бумажном носителе, за исключением оттиска печати и изображения Государственного герба Российской Федерации или герба субъекта Российской Федерации.

Для подписания электронных документов органа власти используются электронно-цифровые подписи (ЭЦП). Используемые средства ЭЦП должны быть сертифицированы в установленном порядке.

При визуализации электронного документа, подписанного электронной подписью, используется отметка об электронной подписи, с соблюдением требований, установленных ГОСТ Р 7.0.97–2016 [3]. Этот стандарт введен в действие с 01 июля 2018 года, и тем самым фактически устранил различие в управлении ЭД в органе власти и в других организациях. В органах власти не используются при оформлении документов реквизиты эмблема, товарный знак (знак обслуживания) и код формы документа. А в обычных организациях не используется реквизит «ссылка на документ». Правила оформления остальных реквизитов абсолютно совпадают.

Состав электронных документов, обращающихся в системе ЭД, определяется руководителем органа власти. Перечень документов, создание, хранение и использование которых может осуществляться исключительно в форме электронных документов, также утверждается руководителем органа власти.

При передаче поступивших электронных документов на рассмотрение руководителю органа власти, направлении электронных документов в структурные подразделения и ответственным исполнителям органа власти, отправке электронных документов и хранении электронных документов вместе с электронными документами передаются (хранятся) их регистрационные данные.

При этом возникает две проблемы: проблема аутентификации документа (разный срок жизни электронной и «живой» подписи) и проблема отсутствия единой модели ЭД (содержимое запоминающего устройства можно интерпретировать как целое число без знака или со знаком, число с плавающей точкой или строкой символов).

Единицей учета электронного документа является электронный документ, зарегистрированный в системе ЭД органа власти.

Требования к информационным системам ЭД органов власти прописывает Приказ Министерства связи и массовых коммуникаций РФ [4]. Эти требования, нацелены на обеспечение работы с электронными документами, их защиту и безопасность.

Исполненные электронные документы систематизируются в дела в соответствии с номенклатурой дел органа власти. При составлении номенклатуры дел указывается, что дело ведется в электронном виде.

Электронные документы после их исполнения подлежат хранению в установленном порядке в органе власти в течение сроков, предусмотренных для аналогичных документов на бумажном носителе.

Единицей учета электронного документа является электронный документ, зарегистрированный в системе ЭД органа власти.

После истечения срока, установленного для хранения электронных дел, на основании акта о выделении их к уничтожению, утверждаемого руководителем органа власти, указанные электронные дела (электронные документы) подлежат уничтожению.

С введением электронного документооборота связан и Перечень обязательных сведений о документах, используемых в целях учета и поиска документов в системах ЭД органов власти, приведенный в приложении к Правилам [5]. Этот Перечень в настоящее время содержит 21 атрибут. В системе ЭД органа власти могут использоваться и дополнительные сведения о документах

Таким образом, управление ЭД в органах власти требует соблюдения требований целого ряда нормативных актов. И важно подчеркнуть, что внедрению систем ЭД обязательно должно предшествовать наведение порядка в «ручном» делопроизводстве.

Реализация государственной программы «Цифровая экономика России» невозможна без информатизации государственной системы ДОУ деятельностью органов власти. Это непосредственно связано с внедрением систем ЭД и требует решения целого комплекса организационных и научно-технических задач как теоретического, так и практического характера.

Библиографический список

1. Концепция формирования в Российской Федерации электронного правительства : [утв. распоряжением Правительства РФ № 632 от 06.05.2008].
2. Рекомендации по подготовке федеральными органами исполнительной власти перечней документов, создание, хранение и использование которых должно осуществляться в форме электронных документов при организации внутренней деятельности : [утв. приказом Росархива № 32 от 29.04.2011].
3. ГОСТ Р 7.0.97–2016. Система стандартов по информатизации, библиотечному и издательскому делу. Организационно-распорядительная документация. Требования к оформлению документов : [утв. приказом Росстандарта № 2004-ст. от 08.12.2016].
4. Об утверждении Требований к информационным системам электронного документооборота федеральных органов исполнительной власти, учитывающих в том числе необходимость обработки посредством данных систем служебной информации ограниченного распространения : приказ Министерства связи и массовых коммуникаций РФ № 221 от 02.09.2011.
5. Правила делопроизводства в федеральных органах исполнительной власти : [утв. постановлением Правительства РФ № 477 от 15.06.2009, с изм. от 07.09.2011, 26.04.2016].

А. А. Корченко¹, Н. К. Жумангалиева²

¹ Национальный авиационный университет, Киев, Украина

² Казахский национальный педагогический университет имени Абая, Алматы, Казахстан

СТРУКТУРНАЯ МОДЕЛЬ СИСТЕМЫ ВЫЯВЛЕНИЯ ВТОРЖЕНИЙ

Расширение воздействий кибератак, направленных на различные ресурсы информационных систем, инициирует задачи построения технических решений и создание специальных средств, способных оставаться эффективными при появлении новых видов угроз с неустановленными или нечетко определенными параметрами. На базе известной методологии построения систем выявления аномалий, порожденных кибератаками, разработана структурная модель системы выявления атак. Она, за счет баз данных кибератак, правил и эталонов, а также модулей формирования текущих значений, α -уровневой номинализации, идентифицирующих термов, уровня аномальности и визуализации, позволяет строить средства, расширяющие функциональные возможности современных систем обнаружения вторжений. Это достигается посредством определения уровня аномального состояния, характерного для воздействия определенного типа кибератак в слабоформализованной нечеткой среде окружения.

Интенсивное развитие информационных систем и технологий привело к увеличению разрушающего программного обеспечения, множество которого направлено на получение конфиденциальной информации, что непосредственно связано с появлением атак типа 0-day и несигнатурных типов кибератак. Также защита от такого рода атак является приоритетной задачей для всех стран.

Большинство современных систем обнаружения вторжений дорогостоящие, имеют закрытый код и требуют квалифицированной настройки под конкретные требования организации и сервисы. На сегодняшний день очень востребованы и необходимы системы выявления вторжений, которые ориентированы на обнаружение аномальных состояний. Они, как правило, содержат профиль нормальной (ненормальной) активности системы и детектируют отклонения от него. Недостатками аномальных систем обнаружения вторжений являются: высокий уровень ложных срабатываний, поскольку любое (даже не опасное) отклонение от нормальной активности спровоцирует реакцию системы; сложность создания обучающей выборки; сложность внедрения и настройки, поскольку стратегия обучения и характер данных для обучающей выборки зависит от специфики вычислительной среды.

Более эффективные в этом отношении являются экспертные подходы, основанные на использовании знаний и опыта специалистов соответствующей предметной области [1].

Исходя из этого построение технических решений и создание специальных средств (например, систем выявления вторжений), позволяющих детектировать ранее неизвестные кибератаки, например, путем контроля текущего состояния нечетко определенных параметров слабоформализованной среды окружения, основанного на экспертных подходах, является актуальной научной задачей.

На базе методологии построения систем выявления аномалий, порожденных кибератаками [2], в основу которой заложен логико-лингвистический подход [1] и кортежная модель [3], а также разработанных методов: формирования лингвистических эталонов; фаз-

зификации параметров на лингвистических эталонах; α -уровневой номинализации нечетких чисел; определения идентифицирующих термов; формирования базовых детекционных правил, построим структурную модель системы выявления кибератак. Она позволит эффективно детектировать в слабоформализованной нечетко определенной среде аномальное состояние за заданный временной промежуток.

Структурная модель системы выявления кибератак [4] содержит согласованные по параметрам базы данных кибератак, правил эталонов, а также модули формирования текущих значений, α -уровневой номинализации, идентифицирующих термов, уровня аномальности и визуализации.

Соответствующая система выявления кибератак (построение которой осуществляется согласно известной методологии [2] посредством 7 этапов) функционирует согласно алгоритму предложенному в [4].

Условно работу структурной модели системы выявления кибератак можно представить двумя процессами:

- 1) процесс инициализации баз данных;
- 2) процесс выявления кибератак.

Процесс инициализации баз данных связан с наполнением (модификацией) баз данных кибератак, баз данных правил баз данных эталонов. При необходимости, на этапе функционирования системы выявления кибератак, указанные базы данных могут подвергаться модификации.

Процесс выявления кибератак CA_i осуществляется за заданный временной промежуток τ_h в каждый момент времени τ_f ($f = \overline{1, \max_\tau}$, где \max_τ – максимальный номер временного промежутка f) на основе множества значений счетчиков сенсоров CS_{ij} (см. (3) в [5]), показатели которых зависят от τ_h ($CS_{ij}^{\tau_h}$), а также эталонных нечетких чисел \tilde{T}_{ijs}^e , которые передаются из базы данных эталонов и поступают в модуль формирования текущих значений, где формируются текущие значения нечетких параметров $\tilde{P}_{ij}^{\tau_f}$ и определяется \max_τ .

Далее с базы данных эталонов и модуля формирования текущих значений соответственно эталонные \tilde{T}_{ijs}^e и текущих $\tilde{P}_{ij}^{\tau_f}$ нечеткие числа поступают в модуль α -уровневой номинализации, где осуществляется их α -уровневая номинализация.

В результате этого с модуля α -уровневой номинализации на вход модуля идентифицирующих термов поступают преобразованные нечеткие числа \tilde{T}_{ijs}^{ep} и $\tilde{P}_{ij}^{\tau_f P}$, где определяются идентифицирующие термы (у которых $s = NUM_{ij}$), отображающие аномальность текущего состояния среды окружения, порожденную определенными кибератаками.

Далее, на основе полученных в модуле α -уровневой номинализации идентифицирующих термов \tilde{T}_{ijs}^{ep} и термов, для которых $s = NUM_{ij}$, поступивших с модуля идентифицирующих термов, а также бинарных решающих функций SF_{ia} (см. (15) в [6]) и идентификаторов аномальности IA_{iu} (см. (5) в [6]), поступающих с баз данных правил, в модуль уровня аномальности формируются подмножества базовых правил DR_i (см. (21) в [6]), посредством которых определяется условное выражение $DR_{iw,s}$, по которому осуществляется выявление i -й кибератаки.

На основании подмножества α -уровневых интервалов AL_{ij}^{le} , межточечных α -уровневых интервалов AL_{ij}^{lp} , а также всех преобразованных \tilde{T}_{ijs}^{ep} и $\tilde{P}_{ij}^{\tau_f P}$, поступивших с модуля α -уровневой номинализации условного выражения $DR_{iw,s}$, поступившего с модуля уровня аномальности, в модуль визуализации графически интерпретируются идентификаторы атакующих действий (отображаемые посредством многомерных (например,

двумерных или трехмерных) опорных областей, например, «Низкий», «Больше низкий чем высокий», «Больше высокий чем низкий», «Высокий», «Предельный» [7]) и фазсифицированные значения текущих параметров $\sim P_{ij}^{x_{fP}}$ относительно лингвистических эталонов T_{ij}^{ep} соответственно.

В работе предложена структурная модель системы выявления кибератак, которая за счет баз данных кибератак, правил и эталонов, а также модулей формирования текущих значений, α -уровневой номинализации, идентифицирующих термов, уровня аномальности и визуализации позволяет строить средства, расширяющие функциональные возможности современных систем обнаружения вторжений, посредством определения уровня аномального состояния, характерного воздействию определенного типа кибератак в слабоформализованной нечеткой среде окружения.

Библиографический список

1. Корченко, А. Г. Построение систем защиты информации на нечетких множествах : Теория и практические решения / А. Г. Корченко. – К. : МК-Пресс, 2006. – 320 с.
2. Корченко, А. Методология построения систем выявления аномалий, порожденных кибератаками / А. Корченко, В. Щербина, Н. Вишневская // Захист інформації. – 2016. – Т. 18, № 1. – С. 30–38.
3. Korchenko, A. The Tupel Model of Basic Components' Set Formation for Cyberattacks / A. Korchenko, K. Warwas, A. Klos-Witkowska // Proceedings of the 2015 IEEE 8th International Conference on «Intelligent Data Acquisition and Advanced Computing Systems : Technology and Applications» (IDAACS'2015) (Warsaw, Poland, September 24–26, 2015). – 2015. – Vol. 1. – P. 478–483.
4. Терейковский, И. А. Система выявления кибератак / И. А. Терейковский, А. Корченко // Безпека інформації. – 2017. – Т. 23, № 3. – С. 176–180.
5. Корченко, А. А. Метод фазсификации параметров на лингвистических эталонах для систем выявления кибератак / А. А. Корченко // Безпека інформації. – 2014. – № 1 (20). – С. 21–28.
6. Карпинский, Н. Метод формирования базовых детекционных правил для систем обнаружения вторжений / Н. Карпинский, А. Корченко, С. Ахметова // Захист інформації. – 2015. – Т. 17, № 4. – С. 312–324.
7. Korchenko, A. A system for identifying anomaly state in informational systems / A. Korchenko, Z. Alimseitova, N. Zhumangaliyeva // VII Inter. University Conference of Students, PhD Students and Young Scientists «Engineer of XXI Century» (Bielsko-Biala, Poland). – 2017. – Vol. 2. – P. 39–48.

В. И. Волчихин, С. А. Зинкин, Н. С. Карамышева

Пензенский государственный университет, Пенза, Россия

ОРГАНИЗАЦИЯ МЕТАКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА ПЛАТФОРМЕ АВТОНОМНЫХ МОБИЛЬНЫХ АГЕНТОВ

Предложена методика организации виртуальной архитектуры агентно-ориентированного метакомпьютера, которую может реализовать единственный агент, выполняя поручение клиента. При этом он может создавать свои клоны и распределять их по физическим узлам вычислительной сети, используя при этом знания о структурных и функциональных связях между объектами – разнообразными ресурсами метакомпьютерной системы. Далее, используя программно реализуемые декларативные и процедурные знания, исходный агент совместно с другими агентами может обрабатывать поступающие данные.

Введение

В последнее время большое внимание уделяется проблемам использования вычислительных сетей как больших вычислительных систем, предназначенных для решения задач, не требующих интенсивного взаимодействия между параллельно исполняемыми частями. Связанная с подобной проблемой технология метакомпьютинга определяется в работах [1, 2] как «использование мощных вычислительных ресурсов, доступных прозрачно посредством телекоммуникационной среды». С другой стороны, существует мнение, что концепция метакомпьютерной технологии не предъявляет особых требований к вычислительным мощностям процессорных ресурсов, в качестве которых могут выступать рабочие станции и персональные компьютеры, подключенные к сети. Поэтому более общим следует считать определение метакомпьютера как виртуального компьютера, динамически организуемого из географически распределенных ресурсов, соединенных сетями передачи данных [3]. В состав метакомпьютерной системы, таким образом, могут входить как простые персональные компьютеры, так и мощные кластерные и массивно-параллельные системы. В настоящей работе рассмотрены проблемы выбора, разработки и интеграции технологий программной реализации сетевых агентно-ориентированных метакомпьютерных систем.

1. Выбор платформы для реализации метакомпьютерных систем

Реализуемое метакомпьютером распределенное приложение должно эволюционировать от известной сетевой архитектуры типа «клиент-сервер» к менее апробированной архитектуре «клиент-агент-сервер» и далее к сложным сетевым архитектурам «клиент-агент – мультиагентная система» и «клиент-агент – агентно-ориентированный метакомпьютер». Рассматриваемый с позиций облачных вычислений, метакомпьютер может представлять один из облачных сервисов (рис. 1). Развитие клиент-серверной сетевой архитектуры агентно-ориентированных метакомпьютеров иллюстрирует рис. 2. Там, где это не противоречит контексту, мы будем рассматривать агентно-ориентированную систему (АОС) как мультиагентную (МАС). Однако понятие агентно-ориентированной системы АОС шире понятия МАС. Например, в работе [4] АОС характеризуется как система, ключевым понятием в которой является абстракция агента.

В данной работе предлагается виртуальная архитектура агентно-ориентированного метакомпьютера, которую может реализовать единственный агент, выполняя поручение клиента, или пользователя, создавая свои клоны и распределяя их по физическим узлам вычислительной сети, используя при этом знания о структурных и функциональных связях между объектами – разнообразными ресурсами метакомпьютерной системы, а затем, используя заданные в программах декларативные и процедурные знания, совместно с другими агентами обрабатывать поступающие данные.

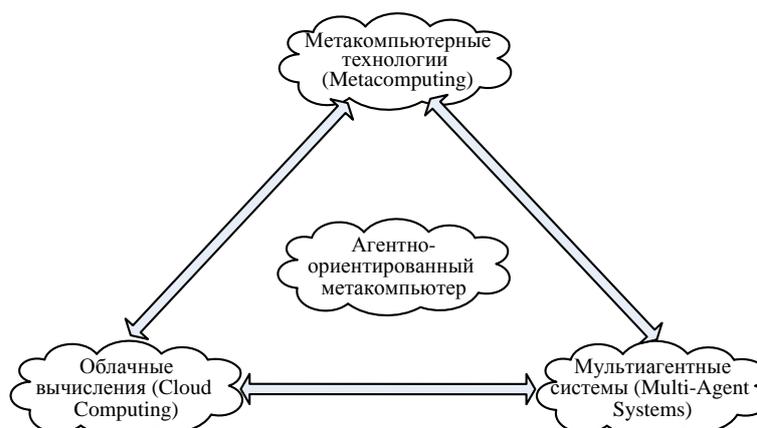


Рис. 1. Интеграция технологий при проектировании агентно-ориентированного метакомпьютера как облачного сервиса

В работе [4] под мультиагентной системой подразумевается система, в которой функционируют несколько агентов, способных взаимодействовать, сотрудничать, догово-

риваться или соперничать. Проблемы декомпозиции, модульности и синхронизации, рассматриваемые в параллельном и распределенном программировании, характерны и для мультиагентных систем. Асинхронное в общем случае взаимодействие агентов удобно формализовать сетями абстрактных машин, узлы которых описываются логико-алгебраическими выражениями.



Рис. 2. Развитие сетевых агентно-ориентированных архитектур: от простой клиент-серверной архитектуры до агентно-ориентированной метакомпьютерной архитектуры облачного сервиса

Целью настоящего исследования является разработка методики абстрактного и структурного проектирования асинхронных систем логического управления для распределенных агентно-ориентированных систем, реализуемых в облачной среде. К таким системам отнесем также и агентно-ориентированные grid-системы.

Под облачными вычислениями подразумевается такая технология обработки данных, при которой компьютерные ресурсы предоставляются пользователям как Интернет-сервисы, или так называемые облачные сервисы. В процессе облачной обработки данных информация хранится на серверах в Интернете и может временно кэшироваться на клиентской стороне [5]. Программное обеспечение, при помощи которого обеспечивается согласованная работа серверов, предоставляющих облачные услуги, относится к классу middleware. Это программное обеспечение осуществляет выделение вычислительных ресурсов для решения задачи, балансировку нагрузки и мониторинг состояния сетевого оборудования, в первую очередь серверов. Распространен такой вид виртуализации сетевых ресурсов, как виртуализация серверов с размещением виртуальных серверов на реальных компьютерах [6].

2. Реализация агентно-ориентированного метакомпьютера как облачного сервиса

Использование свойств гибкости и прозрачности облачных сервисов позволяет создавать на их основе виртуальные среды для распределенной обработки данных. Функциональные возможности облачных сервисов могут быть расширены за счет использования мобильного программного обеспечения, в частности, за счет использования технологий мобильных агентов. Обобщенная модель платформы агентов представлена на рис. 3. Данная модель разработана организацией FIPA (Foundation for Intelligent Physical Agents) для программных агентов и описана в работах [7, 8].

Согласно описанию, агенты в мультиагентной системе функционируют под управлением платформы агентов. Платформой представлены основные службы, которые желательно иметь в любой мультиагентной системе – это службы, обеспечивающие создание и уничтожение агентов, поиск и распознавание агентов, а также коммуникации между агентами. Локальная

служба каталогов позволяет агенту получить сведения о том, какие еще агенты имеются на данной платформе. К службе каталогов могут иметь доступ другие агенты, в том числе агенты, находящиеся на других платформах. Агенты связываются друг с другом, пересылая сообщения и разделяя при работе общее информационное пространство.

Распределенные службы каталогов, каналы связи, организованные при помощи серверов и другие средства платформы агентов позволяют строить сложные структуры, предназначенные для обработки данных. На рис. 4 представлена упрощенная концептуальная схема формирования виртуальной архитектуры метакomпьютера как облачного сервиса.

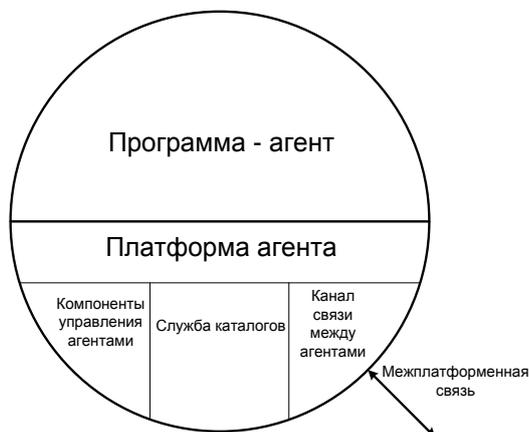


Рис. 3. Обобщенная модель платформы агента

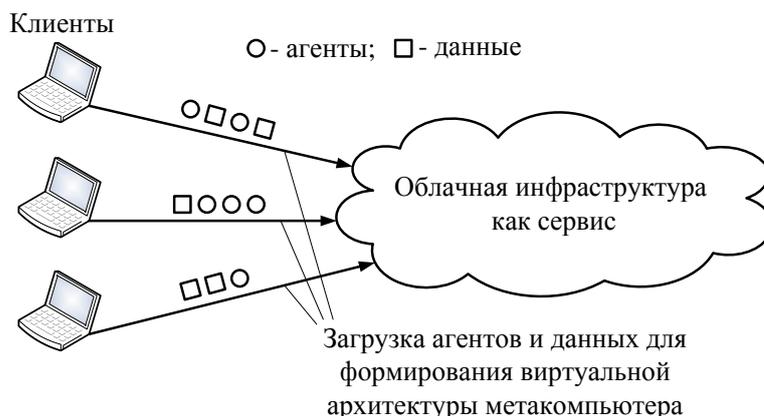


Рис. 4. Реализация метакomпьютера как облачного сервиса: загрузка облачной инфраструктуры

Реальная инфраструктура агентно-ориентированного метакomпьютера формируется путем загрузки на выбранные физические узлы агентов и данных. Агенты разделены на различные категории и обеспечивают функционирование заданных сервисов (приложения как сервиса, управления как сервиса, хранения как сервиса, базы знаний как сервиса и др.). Рисунок 5 иллюстрирует предлагаемую нами концепцию формирования виртуальной архитектуры метакomпьютера как гибридного облачного сервиса «Земля-облако», в которой вычислительные ресурсы пользовательских компьютеров рассматриваются как дополнительные сервисы.

Определены также агенты-супервизоры, агенты-диспетчеры вычислений, агенты-хранители данных, интерфейсные агенты, связанные агенты, агенты-координаторы. Выбор основных типов агентов должен быть основан на анализе агентно-ориентированных технологий. Загрузка агентов и данных, определяющих виртуальную архитектуру метакomпьютера, осуществляется с клиентских компьютеров, которые могут предоставлять свои вы-

числительные ресурсы («землю») другим пользователям посредством облачного сервиса коммуникаций.

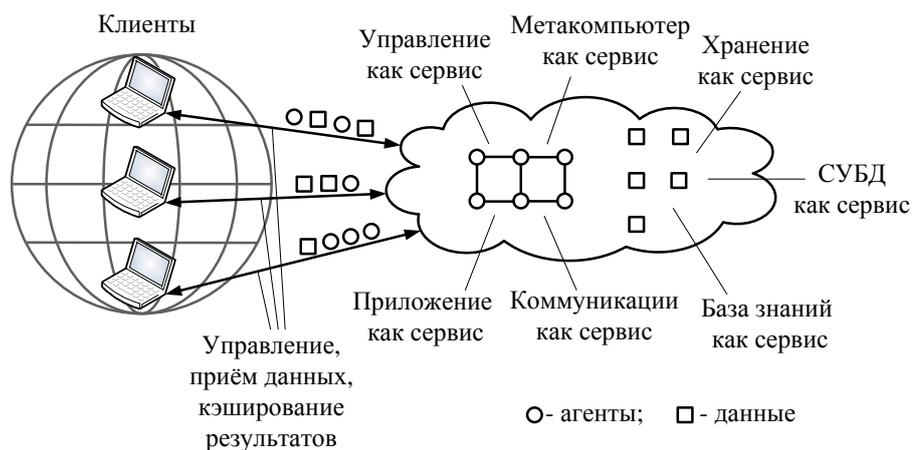


Рис. 5. Реализация метакомпьютера как гибридного облачного сервиса «Земля-облако»

На рис. 6 представлена структура связей в мультиагентной системе обработки данных. Здесь представлены логические связи между пятью мобильными агентами a_1, a_2, \dots, a_5 . Эти связи должны сохраняться на протяжении всего времени решения задачи в мультиагентной системе (агентно-ориентированном метакомпьютере). Каждый агент выполняет одну или несколько функций из набора F_1, F_2, \dots, F_{20} .

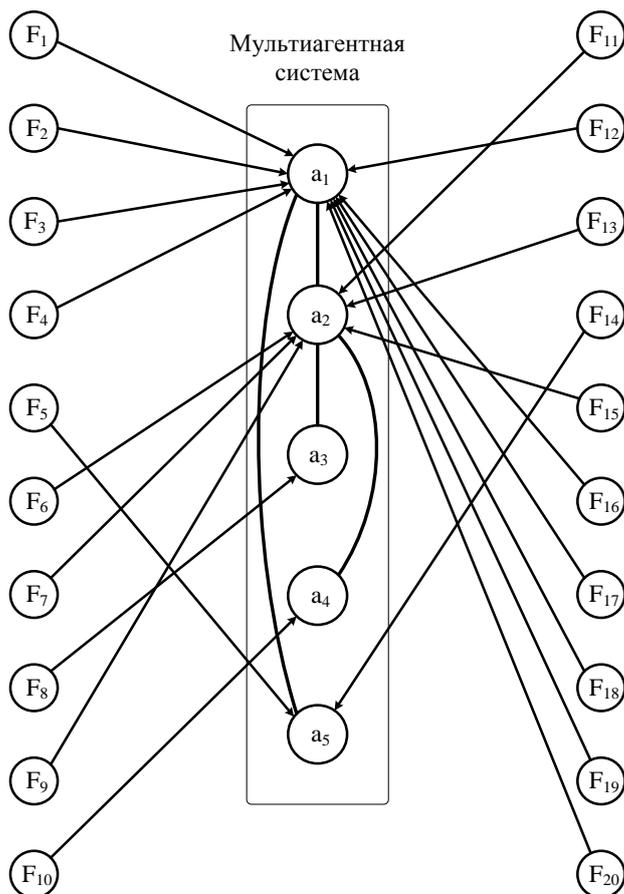


Рис. 6. Логические связи в мультиагентной системе обработки данных

Примеры двух вариантов коллективных пересылок данных между агентами по принципу «каждый с каждым» представлены на рис. 7. Путем обмена сообщениями в распреде-

ленной интеллектуальной системе логического управления удаленные мобильные агенты получают информацию о состоянии центральной базы знаний, на основании этой информации и с помощью программно реализованных абстрактных модулей вносят изменения в свои локальные базы знаний и затем опрашивают результаты центральному стационарному агенту. Последний вносит изменения в центральную базу знаний. Выполняя описанные действия, агенты обмениваются не только управляющими сообщениями, но исходными данными и результатами.

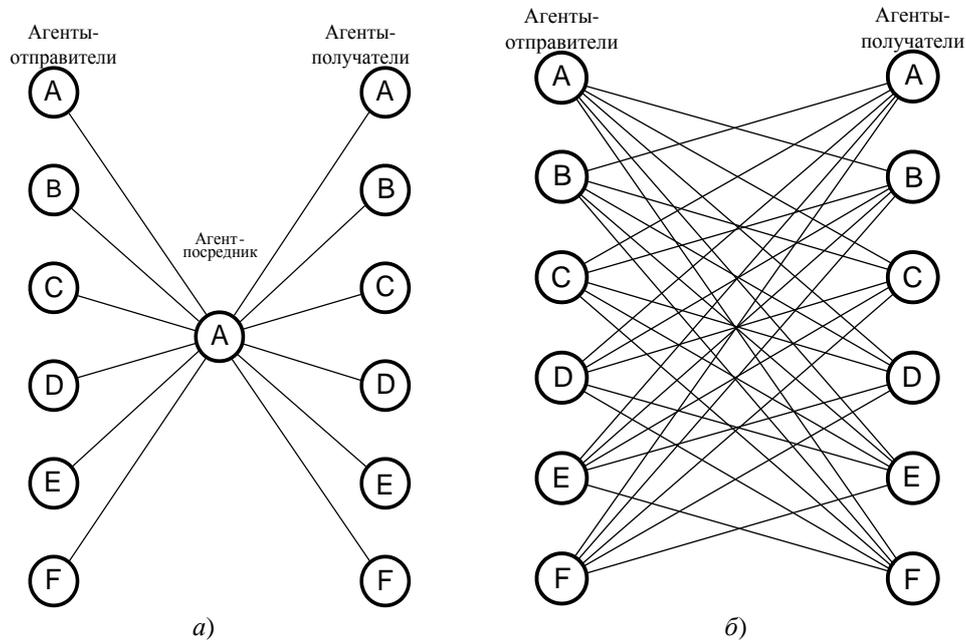


Рис. 7. Коллективные пересылки данных между агентами по принципу «каждый с каждым» через агента-посредника (а) и непосредственно между агентами (б)

3. Концептуальное представление основных событий в агентно-ориентированном метакомпьютере

Рассмотрим представление основных событий в агентно-ориентированных метакомпьютерных системах семантическими сетями и фреймами. В данной работе выбран вариант представления семантических сетей множеством концептуальных графов, представляющих логические формулы. Теоретической основой данного подраздела являются концептуальные графы и их описание на языке первогопорядкового исчисления предикатов, данное в работе [9]. В концептуальных графах имена и аргументы предикатов представлены двумя типами узлов, а дуги графа соединяют имена предикатов с их аргументами. Аргументы представлены прямоугольниками, а предикаты – овалами или кругами. В общем случае предикаты могут иметь несколько аргументов, но чаще используются бинарные предикаты.

Представим графом и логической формулой элементарное событие «Агент_1 передает Сообщение_i Агенту_2». Логическое выражение, описывающее данное событие, имеет следующий вид:

Отправитель(посылка: k, агент: Агент_1)&
Получатель(посылка: k, агент: Агент_2)&
Объект(посылка: k, сообщение: Сообщение_i).

Здесь и далее, если это не противоречит контексту, при переходе логической формулы на следующую строку не будут дублироваться логические связки. Окончание формулы будет, как обычно, обозначаться запятой, точкой с запятой или точкой. Такой стиль принят, например, в основополагающей работе [9]. Концептуальный граф данного события представлен на рис. 8. Здесь значения аргументов типизированы, или, в терминах многооснов-

ного исчисления предикатов, разделены по сортам (основам). Для определения типов использован известный в логике аристотелев подход «через *род (genus)* в *различие (differentia)*» [9]. В нашей интерпретации посылка – «событие (*род*), которое происходит, когда два агента, отправитель и получатель, обеспечивают передачу сообщения через сеть (*различие*)».

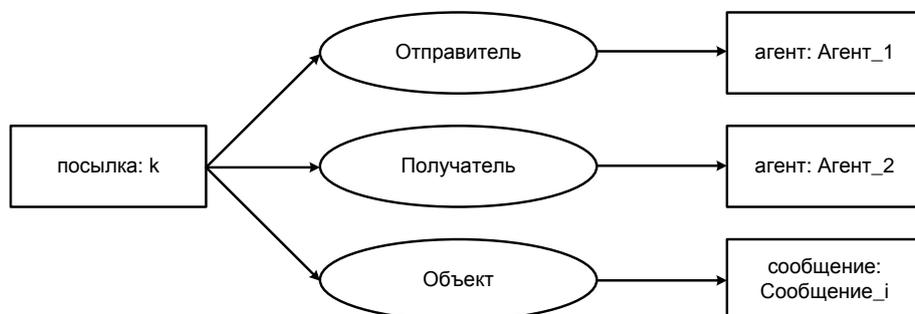


Рис. 8. Концептуальный граф события «Агент_1 передает Сообщение_i Агенту_2»

Формально типы определяются выражениями вида

$$\mathbf{types}(x) \text{ is } F,$$

где s – метка типа события x , F – тело, или различие (*differentia*) метки s , а $s(x)$ – ее род (*genus*).

Приведем пример описания (формальной спецификации) некоторого события x , состоящего в передаче сообщения между двумя агентами $Agent_1$ и $Agent_2$, размещенными на разных узлах:

$$\begin{aligned} &\mathbf{typeTransfer}(x) \text{ is} \\ &Is_a(x, event) \& \\ &Sender(x, Agent_1) \& \\ &Acceptor(x, Agent_2) \& \\ &Object(x, Message_1) \& \\ &Method(x, Message_transfer_protocol) \& \\ &Is_a(Agent_1, activity) \& \\ &Is_a(Agent_2, activity) \& \\ &\neg Equal(Agent_1, Agent_2) \& \\ &Place(Agent_1, Node_1) \& \\ &Place(Agent_2, Node_2) \& \\ &\neg Equal(Node_1, Node_2). \end{aligned}$$

Здесь Is_a – отношение конкретизации, или принадлежности объекта к определенному классу, $Equal$ – отношение равенства, назначение остальных отношений ясно из названий. Действие (событие) $Event_i$, состоящее в передаче сообщения от одного агента к другому, выполняется после модификации предиката $Transfer(x)$:

$$Transfer(Event_i) \leftarrow \square true.$$

Свойства объекта x типа $type_sv$ в общем случае, определяются с помощью правила вывода следующего вида:

$$(\forall x)[Is_a(x, type_s) \supset \&_i Property_i(x, value_i)].$$

Например, свойства события типа $Transfer$ определяются с помощью следующего правила:

$(\forall x)[Is_a(x, type_transfer) \supset Sender(x, Agent_1) \& Acceptor(x, Agent_2) \& Object(x, Message_1)].$

На рис. 9 дано представление предиката $Transfer(x)$ семантической сетью. В данной сети для представления ролей объектов при выполнении события x используются следующие бинарные предикаты: $Sender$ – являться отправителем, $Acceptor$ – являться получателем, $Object$ – являться объектом, $Method$ – применять метод, или протокол, при передаче сообщения, $Place$ – являться сетевым узлом. Значениями аргументов являются следующие имена объектов: $Message_transfer_protocol$ – протокол передачи сообщений, $Node_1$ и $Node_2$ – имена сетевых узлов агента-отправителя и агента-получателя соответственно. Используются следующие классы объектов: $event$ – класс событий, $network_protocol$ – класс сетевых протоколов, $activity$ – класс активных объектов (агентов, процедур), $network_node$ – класс сетевых узлов (то есть физических узлов вычислительной сети).

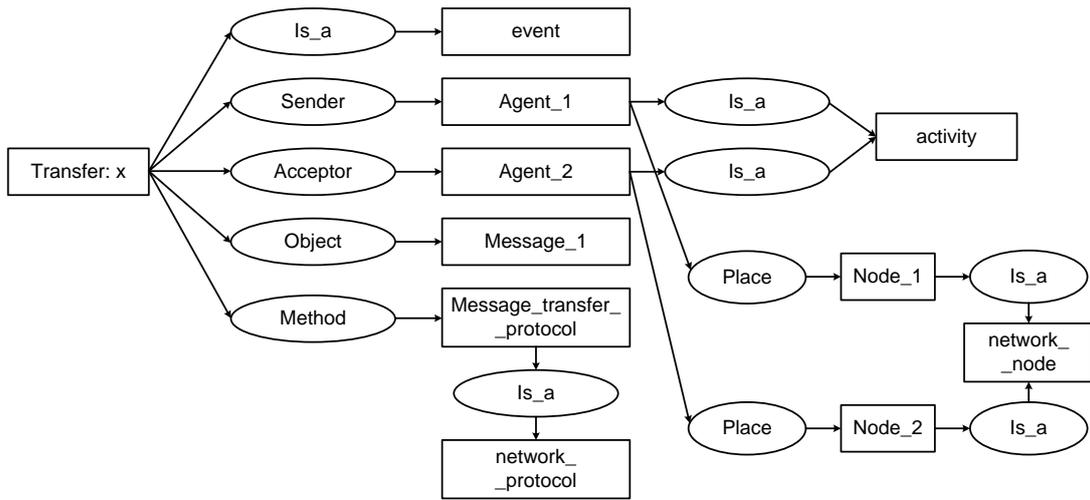


Рис. 9. Представление предиката $Transfer(x)$ семантической сетью

Опишем далее событие y , состоящее в перемещении агента $Agent_1$ на новый узел:

type $Move(y)$ **is**
 $Is_a(y, event) \&$
 $Starting_node(y, Node_1) \&$
 $Receiving_node(y, Node_2) \&$
 $\neg Equal(Node_1, Node_2) \&$
 $Method(y, Agent_transfer_protocol) \&$
 $Object(y, Agent_1) \&$
 $Is_a(Agent, activity) \&$
 $First_place(Agent_1, Node_1) \&$
 $Second_place(Agent_1, Node_2).$

На рис. 10 дано представление предиката $Move(y)$ семантической сетью.

Здесь представлены следующие бинарные предикаты: $Starting_node$ – являться исходным узлом, $Receiving_node$ – являться принимающим узлом, $First_place$ – являться первым, исходным узлом для агента, $Second_place$ – являться вторым, конечным узлом для агента, $Agent$ – являться агентом. Остальные предикаты, объекты и классы были определены выше.

Операции клонирования агента $Agent_1$ и создания его копии $Agent_2$ соответствует событие z , описываемое следующей формальной спецификацией:

type $Clone(z)$ **is**
 $Is_a(z, event) \&$

$Starting_node(z, Node_1) \&$
 $Receiving_node(z, Node_2) \&$
 $\neg Equal(Node_1, Node_2) \&$
 $Method(z, Agent_transfer_protocol) \&$
 $Parent(z, Agent_1) \&$
 $Daughter(z, Agent_2) \&$
 $Is_a(Agent_1, activity) \&$
 $Is_a(Agent_2, activity) \&$
 $Parent_place(Agent_1, Node_1) \&$
 $Daughter_place(Agent_2, Node_2).$

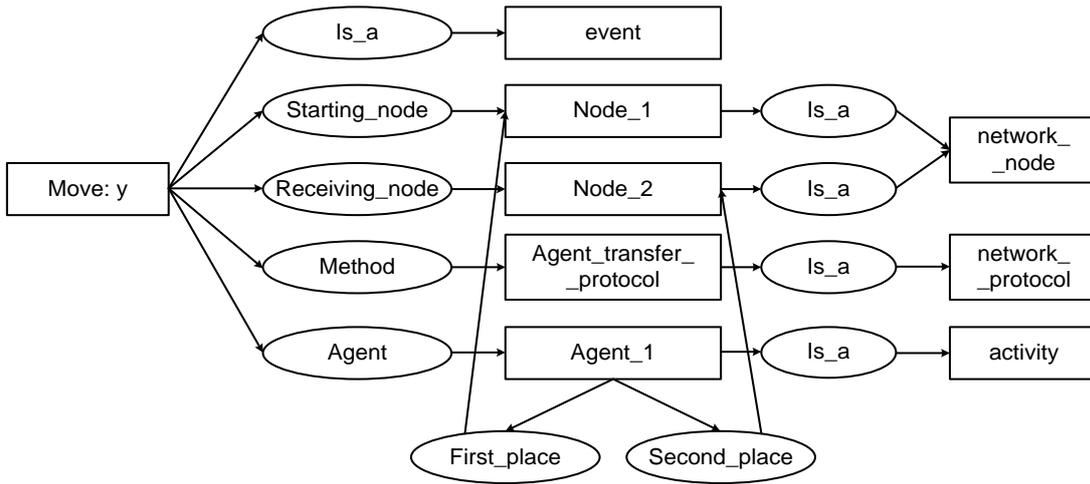


Рис. 10. Представление предиката *Move(y)* семантической сетью

Рисунок 11 иллюстрирует графическое представление предиката *Clon(z)* семантической сетью.

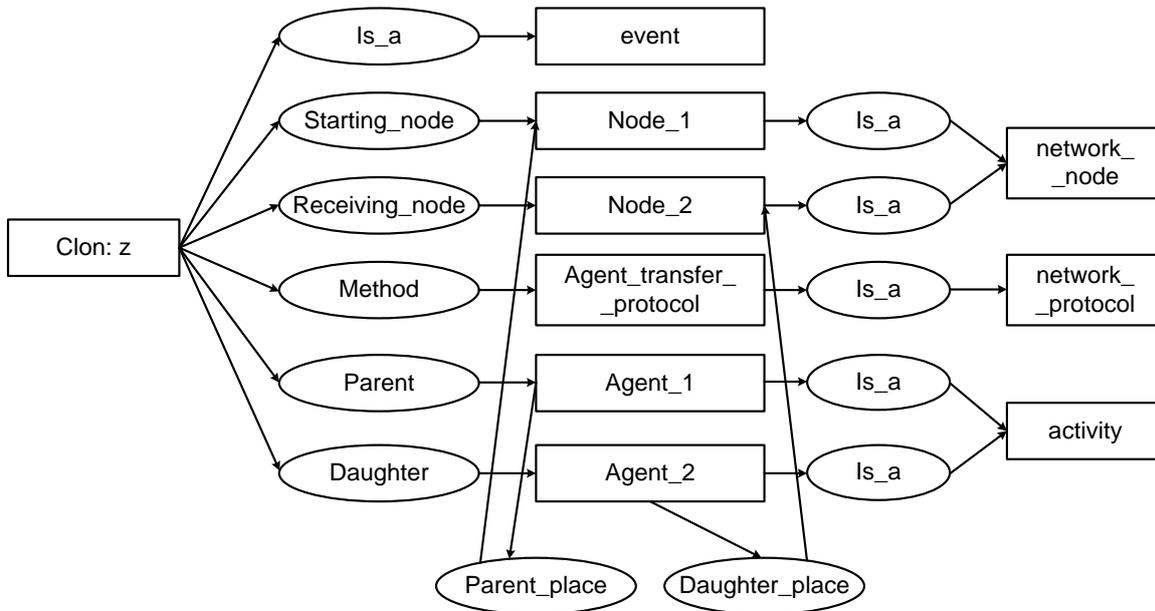


Рис. 11. Представление предиката *Clon(z)* семантической сетью

Здесь использованы следующие бинарные предикаты: *Parent* – являться агентом-“родителем”, *Daughter* – являться “дочерним” агентом (копией или клоном), *Parent_place* – место размещения родителя-агента, *Daughter_place* – место размещения дочернего агента. Остальные предикаты, объекты и классы были описаны выше.

Заключение

1. Проанализированы основные концепции построения метакомпьютерных систем как нового направления в индустрии программирования и вычислительной техники. Анализ показал ряд преимуществ в реализации метакомпьютеров как агентно-ориентированных облачных сервисов, к которым обеспечивается прозрачный доступ пользователей в глобальных или локальных вычислительных сетях:

- упрощается реализация метакомпьютера как виртуальной среды обработки данных, организуемой динамически на основе географически распределенных ресурсов, объединенных сетью передачи данных;

- последующие внедрение и эксплуатация метакомпьютерных систем на промышленном уровне существенно облегчаются, если рассматривать метакомпьютер как один из облачных сервисов, что совместно с применением других технологий виртуализации упрощает создание распределенного программного обеспечения, обеспечивает масштабирование проекта в целом и уменьшает расходы на администрирование;

- за счет специализации агентов и использования свойств мобильности, автономности и способности к взаимодействиям возможна гибкая организация связей между логическими узлами, управления ресурсами, информационного обслуживания, доступа к удаленным данным, запуска и управления распределенными заданиями.

2. Предложено осуществлять реализацию метакомпьютерных систем как агентно-ориентированных облачных сервисов класса «Земля-облако», что позволяет повысить функциональные возможности данных систем, упростить их развертывание, администрирование и восстановление после отказов сетевого оборудования. В предлагаемых системах при необходимости возможно расширить состав вычислительных средств, загружая мобильные агенты с требуемой функциональностью как на серверы, так и на пользовательские компьютеры.

Библиографический список

1. Структура и проблемы развития программного обеспечения среды распределенных вычислений Grid / В. Н. Коваленко, Е. И. Коваленко, Д. А. Корягин, Э. З. Любимский, А. В. Орлов, Е. В. Хухлаев. – М., 2002. – 23 с. – Препринт Института прикладной математики имени М. В. Келдыша Российской академии наук.
2. Catlett, C. Metacomputing / C. Catlett, L. Smarr // Communications of the ACM. –1992. – № 35 (6). – P. 44–52.
3. Коваленко, В. Н. Вычислительная инфраструктура будущего / В. Н. Коваленко, Д. А. Корягин // Открытые системы. – 1999. – № 11–12. – С. 45–52.
4. Wooldridge, M. J. An Introduction to Multiagent Systems / M. J. Wooldridge. – John Wiley & Sons, 2009. – 484 p. – URL: <http://cs.ox.ac.uk>
5. Hewitt, C. ORGs for scalable, robust, privacy-friendly client cloud computing / C. Hewitt // IEEE Internet Computing (September/October). – 2008. – Vol. 12, № 5. – P. 96–99.
6. Dave, P. Introduction to Cloud Computing. Published: 10 April, 2009. – URL: <http://dotnetslackers.com/articles/sql/introduction-to-cloud-computing.aspx>
7. Таненбаум, Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стен. – СПб. : Питер, 2003. – 877 с.
8. FIPASpecifications. – URL: <http://www.fipa.org/specifications/index.html>
9. Тейз, А. Логический подход к искусственному интеллекту: от классической логики к логическому программированию / А. Тейз, П. Грибомон, Ж. Луи. – М. : Мир, 1990. – 429 с.

Мустафа СадекДжафар, С. А. Зинкин

Пензенский государственный университет, Пенза, Россия

О РАЗВИТИИ КОНЦЕПЦИИ «СЕТЬ – ЭТО КОМПЬЮТЕР»

Приведена классификация распределенных вычислительных систем при учете специфики облачных, грид-, кластерных и других видов коммунальных, параллельных и распределенных вычислений. Рекомендовано использовать в качестве базовых гибридные архитектуры, сочетающие положительные свойства облачных, грид- и кластерных распределенных вычислительных систем. Дополнена концепция организации распределенных сетевых вычислений как сервисов, реализуемых по требованию клиента.

Введение

Компания SunMicrosystems, слившаяся со временем с корпорацией Oracle, провозгласила в свое время в качестве девиза концепцию «The Network is the Computer» («сеть – это компьютер»). Получили широкую известность ее ранние технологии, в той или иной степени поддерживающие данную концепцию: Network File System, Java, OpenSolaris, а также основанные на Java-технологиях методы построения распределенных вычислительных систем, например, такие как Remote Method Invocation, Jini, Java Spaces, Java Management Extensions, Jiro, JXTA (Juxtapose). В настоящее время слоган «The Network is the Computer» фактически заменен понятием «Network Computing».

Характерной особенностью современных распределенных вычислительных систем является возможность динамического изменения их функциональной и системной архитектуры, что обеспечивает поддержку реализации концепции «сеть – это компьютер» на основе принципов, характерных для организации коммунальных вычислений. Существует большое число фирм и организаций, занимающихся созданием распределенных сетевых приложений. Интенсивно развиваются современные технологии Cloud Computing, Grid Computing, Global Computing, Community Computing и другие технологии распределенных систем и вычислений, являющиеся, по существу, развитием концепции «The Network is the Computer», что свидетельствует об актуальности выбранного направления исследования в области проектирования распределенных вычислительных систем с облачной сетевой архитектурой, формируемой по требованию клиента.

Концепция «сеть – это компьютер» получила свое дальнейшее развитие в виде парадигмы облачных приложений, обладающих свойствами «мультиа-рендности» и «живой» миграции баз данных. Особый интерес в этой связи представляет разработка гибкого связующего (middleware) программного обеспечения промежуточного уровня для больших переконфигурируемых кластерных серверных систем в рамках реализации средств поддержки концепции «больших данных». Поэтому возник значительный интерес к проектированию реконфигурируемой и параметрически настраиваемой системной и функциональной архитектуры распределенных вычислительных систем. В ряде случаев реализация данного подхода может обеспечить повышение эффективности и снижение стоимости больших программных и аппаратных систем.

1. Классификация распределенных вычислений и систем

Существует ряд классификаций распределенных вычислительных систем, однако многие из них не полны или не в полной степени учитывают современные тенденции разви-

тия распределенных вычислений и систем, рассматриваемых в качестве коммунальных услуг, предоставляемых пользователю.

На рис. 1 представлена классификация распределенных вычислений и реализующих эти вычисления систем. Данная классификация учитывает современные тенденции развития распределенных вычислений и систем, рассматриваемых в качестве коммунальных услуг, предоставляемых пользователю.

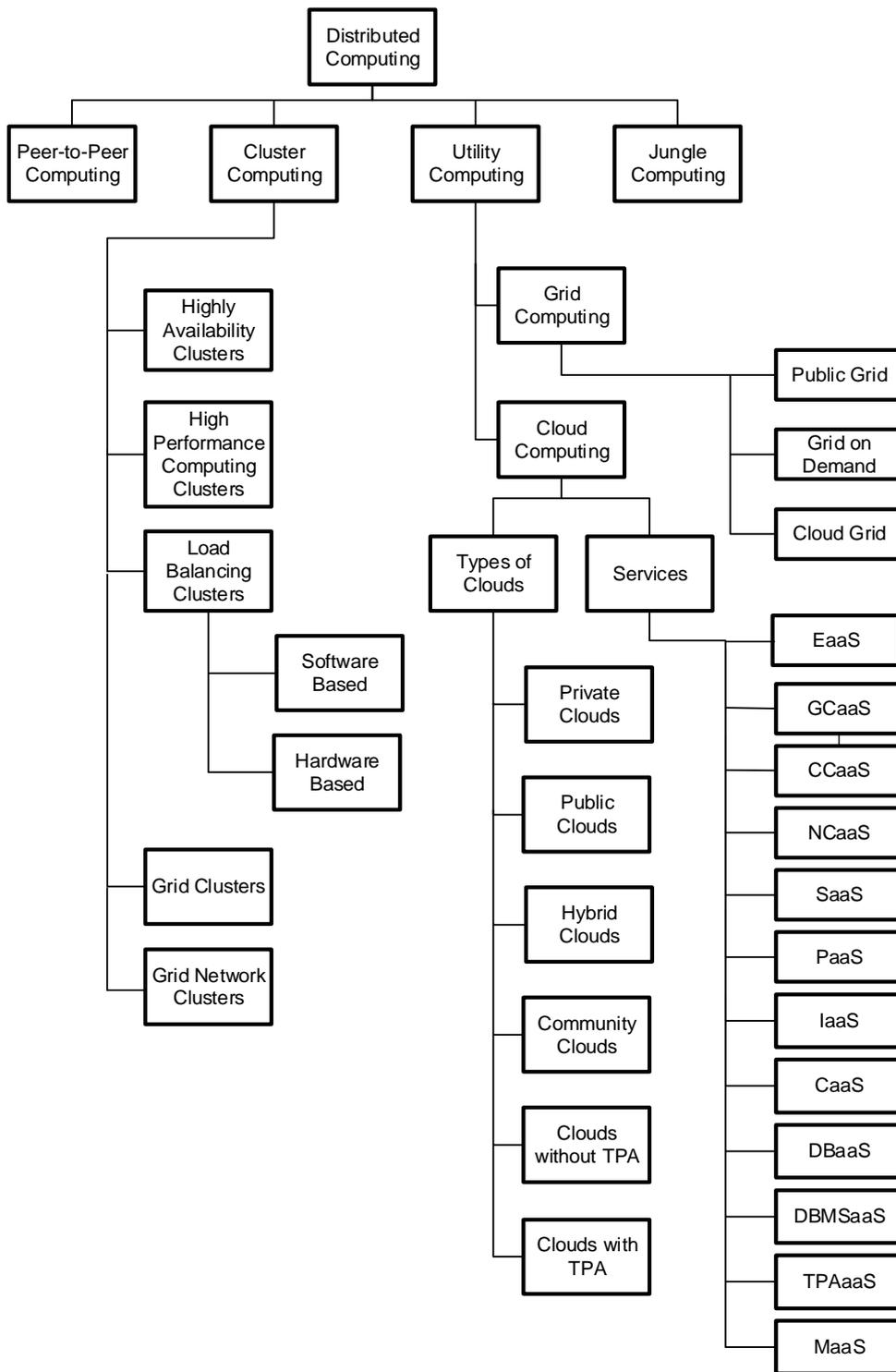


Рис. 1. Классификация распределенных вычислений и систем

Ниже описаны используемые термины и аббревиатуры, а также даны краткие комментарии. Для терминов на английском языке дан перевод. При составлении классификации были приняты во внимание литературные источники [1–12].

- **Distributed Computing (DC)** – вычисления, организуемые в распределенной (возможно, параллельной) вычислительной системе.

- **Peer-to-Peer Computing (PtPC, или P2PC)** – пиринговые вычисления, в реализации которых участвуют, как правило, оконечные компьютеры вычислительной сети, образующие масштабируемую в широких пределах одноранговую децентрализованную вычислительную среду, основанную на равноправии участников.

- **Cluster Computing (CC)** – кластеры (группы) компьютеров, связанные посредством высокоскоростных коммутаторов.

- **Highly Available Clusters (HAC)** – кластеры компьютеров, обеспечивающие высокий уровень надежности и готовности к использованию.

- **High Performance Computing Clusters (HPCC)** – высокопроизводительные кластеры.

- **Load Balancing Clusters (LBC)** – кластеры LBC-компьютеров, обеспечивающих балансировку вычислительной нагрузки.

- **Software Based LBC (SB LBC)** – программно-управляемые кластеры LBC-компьютеров.

- **Hardware Based LBC (HB LBC)** – аппаратно-управляемые кластеры LBC-компьютеров.

- **Grid Clusters (GC)** – кластеры компьютеров на основе высокоскоростных коммутаторов или локальных сетей, образующих грид-систему, т.е. «решетку» компьютеров.

- **Grid Networking Clusters (GNC)** – организуемые в грид-системах виртуальные сетевые кластеры, содержащие в виде узлов отдельные компьютеры физической вычислительной сети (рис. 2).

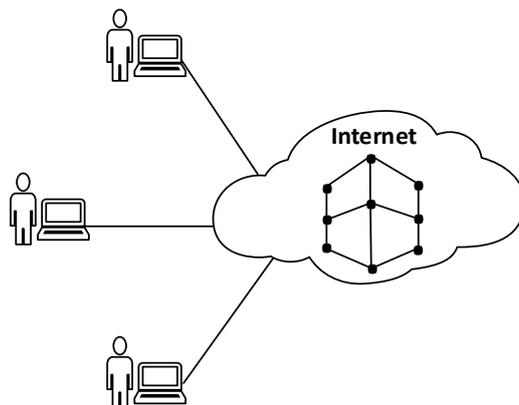


Рис. 2. Концептуальная модель GridNetworkingCluster (виртуального сетевого кластера)

- **Utility Computing (UC)** – вычисления, при реализации которых вычислительные возможности предоставляются как коммунальные услуги.

- **Grid Computing (GC)** – грид-вычисления; для реализации распределенных и, возможно, параллельных вычислений инфраструктура грид-системы образует виртуальный суперкомпьютер, доступный, как правило, нескольким организациям или проектным группам; виртуальный компьютер при этом может содержать как отдельные взаимосвязанные компьютеры, так и целые кластеры компьютеров, связанных высокоскоростными коммутаторами.

- **Public Grid Computing (PGC)** – грид-системы, открытые для коллективного пользования и образуемые свободно предоставляемыми вычислительными и коммуникационными ресурсами.

- **Grid Computing on Demand (GCoD)** – грид-системы типа GCoD; данные системы обычно являются коммерческими приложениями, организуемыми по требованию клиентов.

- CloudGridComputing (CGC) – грид-системы, организуемые в облачных предметно-ориентированных средах (рис. 3).
- CloudComputing (CC) – облачные вычисления, реализуемые как частный случай коммунальных вычислений и предполагающие использование вычислительных ресурсов по типу использования ресурсов сети Интернет.

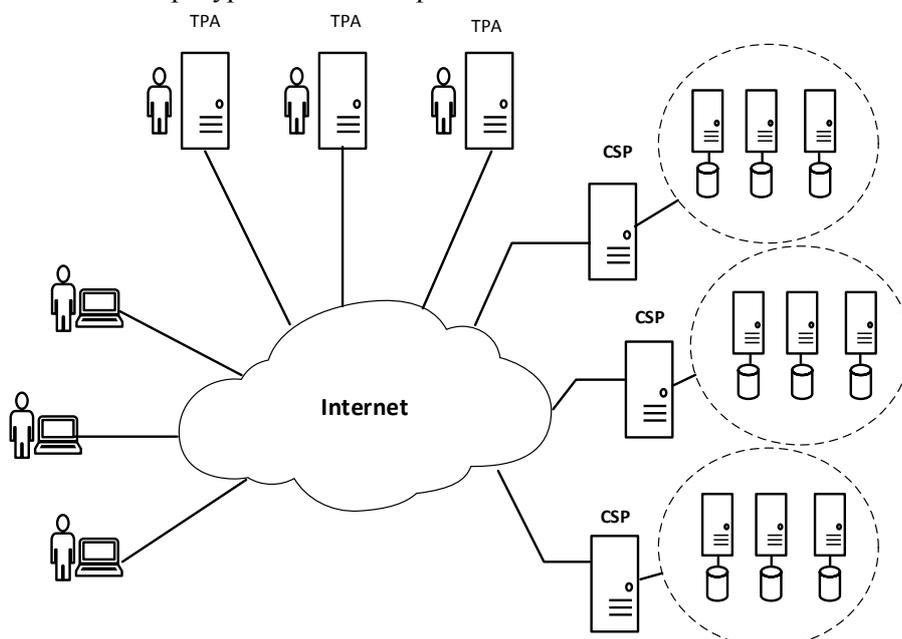


Рис. 3. Концептуальная модель облачной грид-системы CGC (CloudGridComputing) с несколькими сторонними аудиторами TPA (ThirdPartyAuditors) и провайдерами облачных сервисов CSP (CloudServicesProvider)

- TypesofClouds – разновидности облачных вычислений.
- PrivateClouds (PrC) – «приватные, или частные, облака», т.е. облачная среда, реализуемая в вычислительной инфраструктуре, принадлежащей клиенту.
- PublicClouds (PubC) – общедоступная облачная среда.
- HybridClouds (HC) – гибридная облачная среда, включающая как общедоступные, так и частные облачные ресурсы.
- CommunityClouds (ComC) – облачная среда, коллективно используемая сообществом организаций.
- Cloudswithout TPA (CwoTPA) – облачная среда, предоставляющая вычислительные услуги или услуги хранения без участия стороннего аудитора (TPA – ThirdPartyAuditor, аудитор третьей стороны, или сторонний аудитор).
- Cloudswith TPA (CwTPA) – облачная среда, предоставляющая вычислительные услуги или услуги хранения при участии стороннего аудитора TPA.
- Services – сервисы облачной среды.
- Everything as a Service (EaaS) – «всё» как сервис.
- Grid Computing as a Service (GCaaS) – грид-вычисления как сервис.
- Cluster Computing as a Service (CCaaS) – кластерные вычисления как сервис (рис. 4).
- Network Computing as a Service (NCaaS) – сетевые вычисления как сервис.
- Software as a Service (SaaS) – программное обеспечение как сервис.
- Platform as a Service (PaaS) – платформа как сервис.
- Infrastructure as a Service (IaaS) – инфраструктура как сервис.
- Communications as a Service (CaaS) – коммуникации как сервис.
- Data Bases as a Service (DBaaS) – базы данных как сервис.
- Data Base Management System as a Service (DBMSaaS) – системы управления базами данных как сервис.
- Third Party Auditing as a Service (TPAaaS) – сторонний аудит как сервис;

- Monitoring as a Service (MaaS) – мониторинг использования облачной системы как сервис.

- Jungle Computing (JC) – форма организации распределенных вычислений, реализуемая в кластерных, грид- и облачных средах на основе произвольной имеющейся инфраструктуры (Jungle в переводе с английского означает дебри, джунгли) [1, 2, 5].

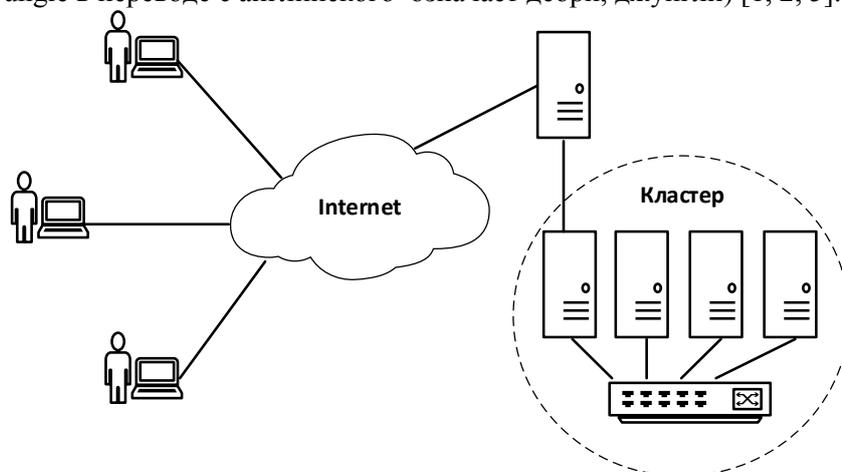


Рис. 4. Концептуальная модель CCaaS – Cluster Computing as a Service (кластерные вычисления как сервис)

Как видно из анализа работ [3, 5], происхождение многих терминов обязано виртуализации процессов и ресурсов в распределенных вычислительных системах.

2. Выбор базовой концептуальной модели для организации функционирования сред, оперативно реализующих распределенные вычисления по требованию клиентов

Распределенные вычислительные системы, реализованные на базе концепции NCaaS, предоставляют клиентам сетевые вычисления как сервис. Реализациям отдельных элементов данной модели в части сохранения целостности, доступности и безопасности данных посвящены работы [13–19]. Концептуально данную модель организации распределенных вычислительных систем иллюстрирует рисунок 5. Данный рисунок из работы [19] модифицирован путем добавления физического или виртуального кластера, организуемого в грид-системе, причем концепция NCaaS расширена до концепции NCaaSSoD – сетевые вычисления как сервис, реализуемый по требованию клиента (Network Computing as a Service on Demand); основные этапы и компоненты, отмеченные на рис. 5 кружками, имеют следующий смысл:

- 1 – загрузка (Upload) в облачное хранилище зашифрованных файлов;
- 2 – загрузка в компьютер стороннего аудитора метаданных для верификации доступа к пользовательским файлам;
- 3 – запрос на верификацию доступа к файлам;
- 4 – вызов процедуры верификации доступа к файлам;
- 5 – ответ от провайдера облачных сервисов;
- 6 – результат верификации для клиента;
- 7 – скачивание (Download) зашифрованных данных из облачного хранилища;
- 8 – создание клиентом метаданных для верификации доступа к файлам;
- 9 – хранение метаданных для верификации доступа к файлам;
- 10 – сохранение зашифрованных файлов;
- 11 – метапланировщик облачных сервисов и сервисов грид-инфраструктуры;
- 12 – коммутатор локальной сети грид-системы (в глобальной сети наряду с коммутаторами используются также маршрутизаторы);
- 13 – хранение зашифрованных пользовательских файлов.

Концептуальная модель NaaS/D, представленная на рис. 5, в целом соответствует принципам организации распределенных и параллельных сетевых вычислений в предметно-ориентированных облачных и грид-средах. Другие модификации концептуальных моделей распределенных вычислений, представленные рис. 2, 3 и 4, также в той или иной степени соответствуют концепции сетевого компьютеринга как сервиса по требованию клиента.

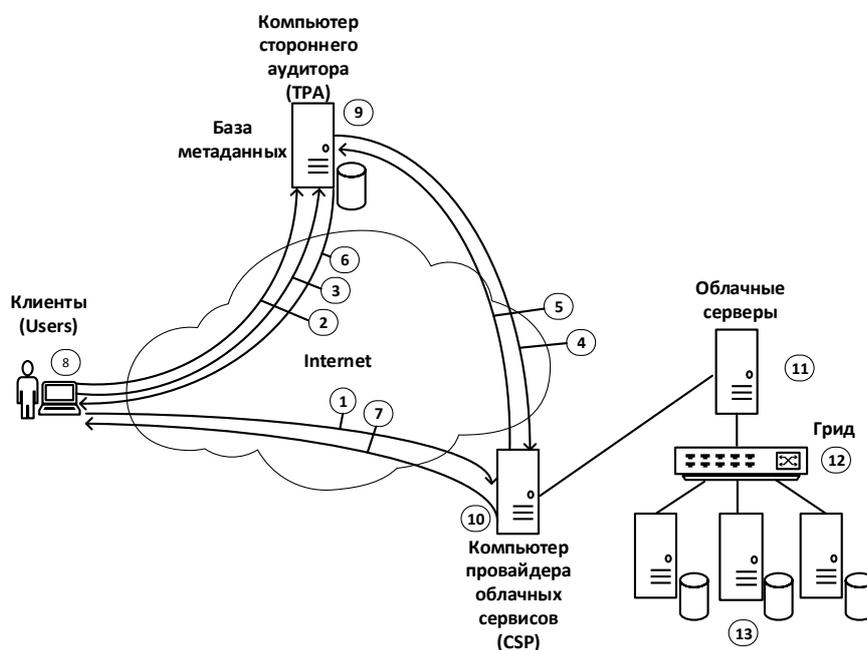


Рис. 5. Концептуальная модель сетевых вычислений как облачногосервиса, реализуемого по требованию клиента (NetworkComputingas a ServiceonDemand) [19, с изм.]

Решения по организации доступа к метаинформации и пользовательским данным в облачных сервисах представлены в следующих работах: основные сведения о функциях стороннего аудитора TPA (ThirdPartyAuditing) и про-вайдера облачных сервисов CSP (CloudServicesProvider) – [19, 20]; сведения о реализации PDP-запросов (ProvableData Possession – доказательство правообладания запрошенными данными) и POR-запросов (ProofofRetrievability – доказательство восстанавливаемости запрошенных данных) на метаинформацию – [20–23].

Вопросы реализации и применения технических и программных средств, функциональность которых настраивается по требованию клиента, на содержательном уровне рассмотрены в работе [24]. В частности, в этой работе предложено при реализации инфраструктуры ODDCI (On-DemandDistributedComputingInfrastructure – распределенная компьютерная инфраструктура по требованию) для распределенных вычислений использовать цифровые телевизионные сети, что позволит обеспечить высокую масштабируемость, гибкость и производительность распределенной компьютерной архитектуры, формируемой по заказу клиента.

Принципы виртуализации распределенных вычислительных систем сформулированы в работах [25–28]. В простейшем случае при виртуализации достаточно реализовать отображение множества логических узлов в сети в множество физических узлов. Коллективные и другие виды обменов информацией в распределенных вычислительных системах можно организовать по принципам, используемым в пиринговых сетях [29–30].

Заключение

Проведена классификация распределенных вычислительных систем и вычислений. Анализ современной литературы показал, что распространение высокоскоростной широкополосной связи между компьютерами в сетях, постоянное увеличение вычислительной

мощности и рост Интернета привели к изменению способов получения вычислительных и информационных услуг. Географически распределенные ресурсы, такие как устройства хранения, источники данных и суперкомпьютеры, взаимосвязаны и могут быть использованы пользователями всего мира как единый ресурс. Отмечена целесообразность использования в качестве базовых гибридных архитектур, сочетающих положительные свойства облачных, грид и кластерных распределенных вычислительных систем.

Библиографический список

1. Kahanwal, B. The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle / B. Kahanwal, T. P. Singh // *International Journal of Latest Research in Science and Technology*. – 2012. – Vol. 1, Iss. 2. – P. 183–187.
2. Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds / Frank J. Seinstra, Jason Maassen, Rob V. van Nieuwpoort, Niels Drost, Timo van Kessel, Ben van Werkhoven, Jacopo Urbani, Cerial Jacobs, Thilo Kielmann, Henri E. Bal // Department of Computer Science, Vrije Universiteit, De Boelelaan 1081A, 1081 HV. – Amsterdam, The Netherlands, 2010. – P. 1–31.
3. Kumar, R. Comparison between Cloud Computing, Grid Computing, Cluster Computing and Virtualization / R. Kumar // *International Journal of Modern Computer Science and Applications (IJMCSA)*. – 2015 (January). – Vol. 3, Iss. 1. – P. 42–47.
4. Baker, M. Cluster computing and applications / M. Baker, A. Apon, R. Buyya, H. Jin // *Encyclopedia of Computer Science and Technology* / ed. A. Kent & J. Williams. – 2002. – P. 87–125. – URL: <http://padabum.com>
5. Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds / Frank J. Seinstra, Jason Maassen, Rob V. van Nieuwpoort, Niels Drost, Timo van Kessel, Ben van Werkhoven, Jacopo Urbani, Cerial Jacobs, Thilo Kielmann, and Henri E. Bal. // *Grids, Clouds and Virtualization* / ed. M. Cafaro, G. Aloisa. – 2011. – XV. – P. 167–198.
6. Mittal, G. A Survey of Current Trends in Distributed, Grid and Cloud Computing / G. Mittal, N. Kesswani, K. Goswami // *International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE)*. – 2013. – Vol. 2, № 3. – P. 1–6.
7. Sadashiv, N. Cluster, Grid and Cloud Computing: A Detailed Comparison / N. Sadashiv, S. M. D. Kumar // *The 6th International Conference on Computer Science & Education (ICCSE 2011) (August 3–5, 2011)*. – Singapore, 2011. – P. 477–482.
8. Jaiswal, U. C. Study and Applications of Cluster Grid and Cloud Computing / U. C. Jaiswal // *International Journal of Engineering Research and Development*. – 2012. – Vol. 3, № 1. – P. 45–50.
9. Kaur, K. A Comparative Analysis: Grid, Cluster and Cloud Computing / K. Kaur, A. K. Rai // *International Journal of Advanced Research in Computer and Communication Engineering*. – 2014. – Vol. 3, № 3. – P. 5730–5734.
10. SamahMawia, Ibrahim Omer. Comparative study between Cluster, Grid, Utility, Cloud and Autonomic computing / Ibrahim Omer SamahMawia, A. Mustafa Amin Babiker, Abdallah ElmahdiAlghali Fatema // *IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)*. – 2014. – Vol. 9, Iss. 6, Ver. III. – P. 61–67.
11. Prathibha, D. A. Issues in adapting cluster, grid and cloud computing for HPC applications / D. A. Prathibha, B. Latha, G. Sumathi // *International Journal of Conceptions on Computing and Information Technology*. – 2014. – Vol. 2, № 1. – P. 12–16.
12. Cloud Computing: Principles, Systems and Applications, series Computer Communications and Networks / B. Amedro, F. Baude, D. Caromel, C. Delbe, I. Filali, F. Huet, E. Mathias, O. Smirnov. – London : Springer, 2010. – P. 163–178.
13. Chourasiya, N. L. Privacy-Preserving Public Auditing for Secure Cloud Storage / N. L. Chourasiya, D. Lature, A. Kumavat, V. Kalaskar // Thaware S. *International Journal of Engineering Research and General Science (IJERGS)*. – 2015. – Vol. 3, № 2. – P. 744–748.
14. Kumarga, N. P. Ensuring Data Integrity in Cloud Computing / N. P. Kumarga, D. Sireesha // *International Journal of Computer Science and Network Security (IJCSNS)*. – 2014. – Vol. 14, № 9. – P. 34–38.
15. Wasnik, N. P. Privacy Preserving Auditing For Cloud Storage / N. P. Wasnik, M. M. Bartere // *International Journal of Science and Research (IJSR)*. – 2015. – Vol. 4, № 5. – P. 2557–2560.
16. Bhagyashri, S. Privacy-Preserving Public Auditing For Secure Cloud Storage / S. Bhagyashri, Y. B. Gurav // *IOSR Journal of Computer Engineering (IOSR-JCE)*. – 2014. – Vol. 16, № 4, Ver. III (Jul.–Aug.). – P. 33–38.
17. Sowparnika, M. Improving data integrity on cloud storage services / M. Sowparnika, R. Dheenadayalu // *International Journal of Engineering Science Invention*. – 2013. – Vol. 2, № 2. – P. 49–55.

18. Kshirsagar1, V. M. A Review on Third Party Auditing by using KERBEROS System for Secure Cloud Storage / V. M. Kshirsagar1, V. S. Gulhane // International Journal of Application or Innovation in Engineering & Management (IJAIEM). – 2014. – Vol. 3, № 3. – P. 426–432.
19. Maher, Alharby. Third-Party based Data Auditing Service (TP-DAS) / Alharby Maher // School of Computer Science. – Manchester, 2015. – P. 1–88.
20. Kumar, B. S. Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification / B. S. Kumar, V. U. Rani, Mustafa Saad // International Journal on Recent and Innovation Trends in Computing and Communication. – 2017. – Vol. 5, № 6. – P. 856–859.
21. Provable data possession at untrusted stores / G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, D. Song // Proc. of CCS'07. – New York, NY, USA: ACM, 2007. – P. 598–609.
22. Juels, A. Pors: proofs of retrievability for large files / A. Juels, D. S. Kaliski // Proc. of CCS'07. – New York, NY, USA: ACM, 2007. – P. 584–597.
23. Bowers, K. D. Proofs of retrievability: Theory and implementation / K. D. Bowers, A. Juels, A. Oprea // Cryptology ePrint Archive, Report 2008/175. – Heidelberg, 2008. – 24 p.
24. Costa, R. OddCI: On-demand distributed Computing Infrastructure / R. Costa, F. Brasileiro, G. L. Filho, D. M. Sousa // Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers (MTAGS'09) (November 16th, 2009). – Portland, Oregon, USA, 2009. – P. 1–10.
25. Distributed Virtual Machines: A System Architecture for Network Computing / E. G. Sirer, R. Grimm, A. J. Gregory, N. Anderson, B. N. Bershad. – URL: <http://www.cs.cornell.edu/people/egs/papers/kimera-tr98-09-01.pdf> (дата доступа: 10.05.2018).
26. Kaur, P. Distributed and Cloud Computing Architecture / P. Kaur, R. Rani // Imperial Journal of Interdisciplinary Research (IJIR). – 2017. – Vol. 3, Iss. 2. – P. 1119–1123.
27. Rabenseifner, R. Hybrid MPI and OpenMP Parallel Programming / R. Rabenseifner, G. Hager, G. Jost, R. Keller // Recent Advances in Parallel Virtual Machine and Message Passing Interface. Euro PVM/MPI 2006. Lecture Notes in Computer Science. – Berlin, Heidelberg, 2006. – Vol. 4192. – P. 11.
28. Venkata, M. G. MPI/CTP: A Reconfigurable MPI for HPC Applications / M. G. Venkata, P. G. Bridges // Recent Advances in Parallel Virtual Machine and Message Passing Interface. EuroPVM/MPI 2006. Lecture Notes in Computer Science. – Berlin, Heidelberg, 2006. – Vol. 4192. – P. 96–104.
29. Chen, Rui. Modeling of UDP Hole Punching in P2P Network Using Petri Net / Rui Chen, Xiao-jing Meng // International Proceedings of Economics Development and Research. – 2012. – Vol. 49. – P. 150–154.
30. Park, H. Peer-to-Peer Networks: Protocols, Cooperation and Competition / H. Park, R. I. Ratzin, M. Schaar // Source Title: Streaming Media Architectures, Techniques, and Applications: Recent Advances, 2011. – 33 p. – URL: <http://ebookey.org>

А. А. Яужев

Пензенский государственный университет, Пенза, Россия

ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ВСТРАИВАЕМЫХ СИСТЕМ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРОВ

Применение программных и аппаратных возможностей микроконтроллеров и специализированных микросхем для реализации защиты программного обеспечения и обрабатываемых данных встраиваемых систем методами контроля целостности, шифрования и согласования ключей.

Встраиваемые системы применяются для решения специализированных задач и зачастую построены с использованием компонентов, обладающих наибольшим соотношением производительности к стоимости. К таким компонентам относятся восьмьбитные микроконтроллеры [1] (AVR, STM8) и 32-битные контроллеры на основе ядер ARM (M4, M3, M0) – большинство из них имеют существенные аппаратные ограничения касательно объема доступной постоянной и оперативной памяти, что приводит к ограничению функциональности исполняемого микроконтроллером программного обеспечения (ПО). ПО встраиваемых систем в общем случае реализует только критически необходимые функции,

игнорируя избыточное кодирование данных и кода в памяти, контроль целостности и шифрование данных при передаче и хранении. Для защиты интеллектуальной собственности, а также обеспечения безотказной работы системы при риске внедрения сторонних компонентов могут использоваться некоторые аппаратные возможности контроллеров или специализированные микросхемы.

К встраиваемым системам, как и к любым другим, могут быть применены следующие методы защиты:

- контроль целостности исполняемого кода;
- шифрование кода и данных в постоянной памяти;
- шифрование данных при взаимодействии с внешними системами.

Методы контроля целостности позволяют обеспечить соответствие исполняемого кода программы эталонному образцу ПО, подготовленному производителем встраиваемой системы, а также подтверждает неизменность обрабатываемых системой данных. Таким образом, предотвращается исполнение произвольного кода в микроконтроллере при нормальных условиях работы системы. Контроль целостности для систем, не предусматривающих обновление ПО в процессе эксплуатации и поддержки, может быть реализован с применением однократно программируемой памяти (ОТР) в качестве основного хранилища исполняемого кода. Для систем, допускающих обновление ПО, необходимо применить к машинному коду программы некоторую хеш-функцию в сочетании с функцией аутентификации (симметричный алгоритм) или подписи (асимметричный алгоритм). Оба подхода требуют использования ОТР памяти для хранения ключа проверки контрольной последовательности, а в случае с симметричным алгоритмом требуется аппаратная поддержка алгоритма аутентификации, так как используемый ключ не должен быть доступен для чтения извне. При модификации ПО производитель выполняет обновление контрольной последовательности для сохранения ее соответствия исполняемому коду программы: это соответствие проверяется в процессе запуска и/или работы системы. Эффективность данного подхода напрямую зависит от сохранности ключа, используемого производителем для формирования контрольных последовательностей.

Применение контроля целостности предотвращает модификацию исполняемого ПО, но не защищает от создания аналогичной копии устройства с переносом кода и методов обратной разработки. Для защиты от такого рода угроз может быть использовано шифрование исполняемого кода при его хранении в постоянной памяти с расшифровкой перед его исполнением, однако такой подход существенно ограничен объемом доступной памяти и производительностью микроконтроллера. При использовании отдельных микросхем для шифрования кода производительность ограничивается скоростью обмена данными между микросхемами и издержками протокола их взаимодействия.

Защита от клонирования данных, хранимых в постоянной памяти, может быть реализована с применением уникальных для каждой системы ключей шифрования, сформированных генератором случайных чисел или на основе некоторых постоянных признаков системы (серийных номеров микроконтроллера, памяти или монотонных счетчиков).

Для защиты передаваемых данных от подделки или имитации может использоваться алгоритм согласования общего ключа шифрования данных с использованием взаимной аутентификации сторон или без нее. Для аутентификации обе стороны должны иметь публичные ключи, подписанные ключом общей доверенной стороны (производитель системы). Общий ключ, полученный при согласовании, может использоваться для контроля целостности, шифрования данных или того и другого одновременно (при реализации GCM [2]). Использование случайного числа при формировании общего секрета позволяет получать уникальные общие ключи для каждого согласования.

Все вышеперечисленные методы защиты требуют наличия той или иной аппаратной возможности для их реализации: от наличия однократно программируемой памяти до аппаратной реализации хранилища ключей и стандартных криптографических алгоритмов. Некоторые микроконтроллеры имеют встроенные аппаратные блоки, реализующие криптографические функции (SAM4L, SAML11, LPC1830 и др.), для остальных случаев возможно применение специализированных микросхем, реализующих аналогичный набор функций (CryptoAuthentication).

Микросхемы Microchip CryptoAuthentication реализуют стандартные криптографические алгоритмы AES, SHA и ECDSA/DH, и поставляются в виде микросхем постоянной памяти с областью конфигурации специальных функций, включая:

- генерацию и хранение криптографических ключей без возможности их извлечения;
- алгоритм вызов-ответ на основе алгоритма SHA-256;
- алгоритм безопасной загрузки с блокировкой криптографических функций при неудаче (SHA-256 + ECDSA);
- электронная подпись произвольных данных (ECDSA).

Микросхема АТЕСС608 способна выполнять все вышеперечисленные функции и позволяет реализовать все представленные методы обеспечения безопасности встраиваемых систем. Для взаимодействия используется протокол I2C, что позволяет использовать ее в различных системах, включая аппаратные средства аутентификации пользователей [3].

Библиографический список

1. Баранов, В. И. Применение микроконтроллеров AVR. Схемы. Алгоритмы. Программы / В. И. Баранов. – М. : Додэка XXI, 2006. – С. 288.
2. The Galois/Counter Mode of Operation (GCM) / D. McGrew, J. Viega. – URL: <https://eprint.iacr.org/2004/193.pdf> (дата обращения: 17.09.2018).
3. Яужев, А. А. Комбинированное средство аутентификации / А. А. Яужев // Новые информационные технологии и системы : сб. науч. ст. XIV Междунар. науч.-техн. конф. – Пенза : Изд-во ПГУ, 2017. – С. 179–181.

А. О. Семенов

Пензенский государственный университет, Пенза, Россия

О РАСПРЕДЕЛЕНИИ ПОЛОСЫ ПРОПУСКАНИЯ В РАМКАХ КОММУТАТОРА НА БАЗЕ СТОХАСТИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ

Объектом исследования являются современные телекоммуникационные устройства (коммутаторы). Предметом исследования являются системы диспетчеризации современных телекоммуникационных устройств. Цель работы – создание дисциплины диспетчеризации, удовлетворяющей требованиям обеспечения качества обслуживания.

Введение. Настоящая работа посвящена оценке эффективности алгоритмов диспетчеризации очередей средствами имитационного моделирования с использованием пакета CPNTools, который основан на применении математического аппарата цветных иерархических сетей Петри [1, 2, 3, 4].

Структура сети Петри, моделирующей алгоритм *St1*, основанного на принципах стохастического выбора очереди, согласно которому: правило выбора очереди – псевдослучайное равновероятное, выбор осуществляется только среди непустых очередей, производимые выборки возвратны; номинальные объемы обслуживания для каждой очереди различны и пропорциональны выделяемым долям полосы пропускания.

В сеть Петри, моделирующую алгоритм *St1* было введено два условных типа маркеров:

1. «Информационные» маркеры, продвижение которых имитирует обработку кадра в коммутаторе:

Множество цветов *frm* отражает структуру передаваемого кадра Ethernet и анализируется на очередном этапе обработки кадра в коммутаторе.

Множество цветов *frame* представляет однонаправленный канал, который может быть либо занят передачей кадра (*frm*), либо свободен (*avail*), что позволяет выявлять и обрабатывать события, связанные с наличием или отсутствием кадра.

«Управляющие» маркеры, цвет которых отображает состояние обработки потока кадров в коммутаторе (размер очереди, количество переданных данных, признак очистки очереди и др.).

Задержки коммутации и сериализации коммутатора задаются на исходящем переходе выходного порта. За единицу модельного времени берется время передачи одного бита в входном/выходном портах.

В состав всех представленных моделей стохастических диспетчеров входят:

○ Генератор трафика (*SetTraf*) – блок, отвечающий за создание и передачу последующим блокам информации в формате, который можно представить в виде кортежа (цвета *frame*)

$$(number, src, dst, qos, szfrm, delay) \quad (1),$$

где *number* – номер кадра; *src* – идентификатор отправителя; *dst* – идентификатор получателя; *qos* – идентификатор класса обслуживания; *szfrm* – размер кадра; *delay* – время (задержка) начала формирования кадра.

3. Входной буфер (*Bufferin*) – позиция-буфер, содержащая накопленные кадры, ещё не прошедшие через процедуру классификации.

4. Буфер выходного порта (*Buffer*) – блок, отвечающий за последовательную классификацию, хранение и продвижение кадров разнородного трафика к выходному каналу согласно заранее определённым правилам диспетчеризации.

5. Выходной буфер (*Bufferout*) – позиция-буфер, содержащая кадры, отобранные диспетчером и направляемые в выходной канал.

6. Узел передачи кадра в канал связи (*OutPort*).

7. Позиция-флаг свободного канала связи (*OutPortFree*).

8. Позиция-агрегатор кадров поступивших в канал связи (*output*).

В свою очередь, в состав блока *Buffer* всех представленных моделей стохастического диспетчера входят:

1) Входная связующая позиция подсети (*BufferIn*).

2) Классификатор кадров (*Classifier*).

3) *Queue1In...Queue4In* – позиции, содержащие кадры, прошедшие процедуру классификации и направляемые в очереди *Queue1...Queue5*.

4) Подсети очередей, содержащие маркеры кадров и очередей, ожидающих обслуживания (*Queue1...Queue4* и *Sizequeue1out...Sizequeue3out*) в виде двух кортежей (цветов *frame*, см. выше, и *QUEUE_PROPERTIES*)

$$(n, sz, nIsNtEmpty, nBytesTransmitted), \quad (2)$$

где *n* – идентификатор очереди, соответствующей набору классов обслуживания, *sz* – длина очереди, *nIsNtEmpty* – флаг, отображающий наличие кадра(ов) в очереди, *nBytesTransmitted* – примерное количество переданных в канал связи бит по соответствующему классу обслуживания.

5) *bend1, ... bend4* – подсети, выделяющие полосу пропускания для трафика каждой очереди.

6) Выходная связующая позиция подсети (*OutPortFree*).

7) Позиция *Cycle*, содержащая инструкции для блока *rand_scan* в виде кортежа (цвет *QUEUE_RESTS*)

$$(nGen, nCycleScan), \quad (3)$$

где *nGen* – номер очереди, предлагаемой к объективному процессу выборки кадра, *nCycleScan* – номер цикла (фаза) обслуживания выборки.

8) Входная связующая позиция подсети (*Output*) и другие блоки.

В рамках реализации блока *rand_scan* осуществляется генерация псевдослучайного числа $nRand$, согласно равномерному распределению. Каждой j -той очереди ставится в соответствие числовое значение $nCrit_j$: ноль или единица, для пустых и непустых очередей, соответственно, причём

$$nSummCrit_{j+1} = nSummCrit_j + nCrit_{j+1}. \quad (4)$$

Затем числа вычисляется разность $nRand$ и $nSummCrit_j$ для каждой j -й очереди. Выбирается на обслуживание очередь с меньшим идентификационным номером, удовлетворяющая требованию знакоотрицательности вышеназванной разности.

Структура сети Петри, моделирующей алгоритм *St3*, основанного на принципах стохастического выбора очереди, согласно которому: сеансы обслуживания очередей разделены на чередующиеся подсеансы первого типа (циклического обслуживания) и второго типа (стохастического обслуживания), причём номинальные объёмы выборок первого типа различны и пропорциональны выделяемым долям полосы пропускания, причём сеанс обслуживания согласно выборке первого типа делится на равные по длительности подсеансы обслуживания, прерываясь на сеансы обслуживания согласно выборке второго типа, номинальные объёмы обслуживания второго типа определяются величиной длины первого кадра в выбранной наугад очереди; правило выборок первого типа – циклическое (сходно с алгоритмом *DRR*), правило выборок второго типа – псевдослучайное, вероятности каждой из выборок одинаковы, производимые выборки возвратны; сходно со структурой сети Петри, моделирующей алгоритм *St1* за исключением некоторых функциональных особенностей. Представленные изменения заложены в тело функции *fun ChooseBand(i, n, sz, k, szfrm, nCycleNo, cnNoOfCycles)*, осуществляющей разделение большого сеанса на подсеансы путём вызова функции.

В состав подсети *rand_scan* (далее – устройство стохастической диспетчеризации в составе коммутатора или маршрутизатора) очередей сетевого коммутатора с поддержкой качества обслуживания, содержит: генератор псевдослучайных чисел; узел хранения постоянных величин (ПЗУ); узел подсчёта вероятностей выбора очередей на обслуживание Узел сравнения величин; узел подсчёта номера обслуживаемой очереди.

Выводы:

9. Предложено решение, реализующее алгоритмы динамической стохастической диспетчеризации.

10. Проведено моделирование представленных алгоритмов. Как видно из представленных данных, для стохастических алгоритмов зафиксированы вариации средних значений задержек, аналогичные неклассическому алгоритму *DRR-TSS*, и низкие средние значения вариаций задержек относительно классического циклического алгоритма на наборе несогласованных данных. Это говорит о перераспределении неиспользуемой полосы пропускания таким образом, чтобы снизить задержку в перегружаемых очередях за счет наиболее недогруженных.

Библиографический список

1. Кучерявый, Е. А. Управление трафиком и качество обслуживания в сети Интернет / Е. А. Кучерявый. – СПб. : Наука и Техника, 2004. – С. 336.
2. Jensen, K. Coloured Petri Nets: modelling and validation of concurrent systems / K. Jensen, L. Kristensen. – Berlin : Springer Science & Business Media, 2009. – P. 125.
3. Barreiros, M. QoS-enabled networks tools and foundations / M. Barreiros, P. Lundqvist. – John Wiley & Sons, 2016. – 254 p. – URL: <http://certcollection.org>
4. Кизилов, Е. А. Моделирование адаптивной диспетчеризации очередей в коммутаторе с поддержкой QoS / Е. А. Кизилов, Н. Н. Коннов, Д. В. Патунин // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 170–183.

Д. А. Замятин, В. В. Кольга

Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева (СибГУ), Красноярск, Россия

ОСНОВНЫЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ ОРБИТАЛЬНОГО ЦЕНТРА ХРАНЕНИЯ И ПЕРЕДАЧИ ДАННЫХ

Описаны особенности проектирования космического аппарата, выполняющего роль узла компьютерной сети.

Космическая платформа, выполняющая роль узла орбитальной компьютерной сети, строится по модульному принципу, сходному со структурой орбитальных станций.

Основную массу космической платформы составляют серверные модули – крупные контейнеры с размещённым на них компьютерным оборудованием. Именно они являются системой хранения и оперативной обработки информации, орбитальным DATA-центром.

Необходимость использования модульного принципа обусловлена спецификой назначения космической платформы.

На рис. 1 показана трёхмерная геометрическая компьютерная модель, в упрощённом виде демонстрирующая один из вариантов схем сборки модулей космической платформы в единое целое [1].

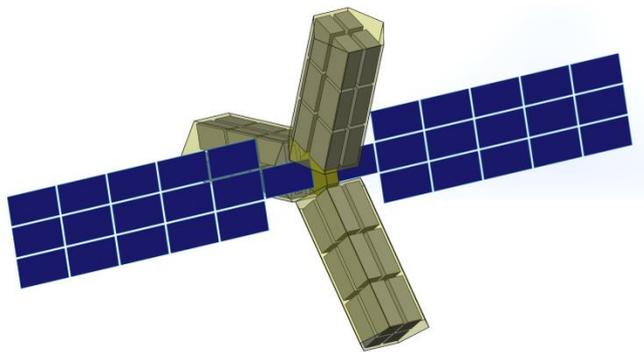


Рис. 1. Вариант схемы сборки космической платформы с креплением модулей хранения информации вокруг модуля энергообеспечения (модули связи не показаны)

Мощное серверное оборудование потребляет большое количество электрической энергии в единицу времени, а габариты модулей хранения информации практически полностью используются для размещения компьютерного оборудования, в связи с чем электроснабжение комплекса должно обеспечиваться отдельным модулем энергообеспечения, что позволит создать достаточную площадь солнечных батарей для питания серверного оборудования электрической энергией.

Модуль хранения и обработки информации представляет собой закрытый контейнер, имеющий форму правильной шестиугольной призмы высотой 7,8 м с диаметром описанной окружности 3,6 м. На каркасе контейнера крепятся блоки сетевых хранилищ. Согласно предварительным расчётам, требования к массе и габаритам контейнера при использовании современного оборудования позволят обеспечить суммарный объём хранимой информации до 270 петабайт. Учитывая, что орбитальные DATA-центры предназначены не для длительного хранения информации, а для её кэширования при передаче, такого объёма сетевых хранилищ предположительно должно хватать для выполнения задач сети в полной мере. Позднее, с повышением плотности данных на носителях информации будущих поколений, орбитальные DATA-центры будут подвергаться обновлениям аппаратного обеспечения [2, 3].

Потребляемая сетевым оборудованием мощность электроэнергии по предварительным расчётам, должна составлять приблизительно 480 кВт, что подтверждает, с одной стороны, целесообразность использования отдельного модуля космической платформы, служащего для электроснабжения сетевых хранилищ, с другой – техническую возможность

реализации концепции без дополнительных затруднений, касающихся проблем энергообеспечения.

Для определения указанных характеристик был проведён поиск и анализ необходимой информации из открытых источников о современных сетевых хранилищах и выполнен краткий расчёт представляющих интерес характеристик в программе Microsoft Office Excel 2007. В табл. 1 приведён расчёт характеристик для одного модуля хранения и обработки информации [3, 4, 5].

Таблица 1

| | Габариты, мм | | Количество ячеек | |
|---------------------------|--------------|------------|------------------|--------|
| | Ячейки | Контейнера | В ряду | Общее |
| Высота | 200 | 7800 | 32 | 1920 |
| Ширина | 200 | 3000 | 10 | |
| Глубина | 200 | 1800 | 6 | |
| Масса, кг | 6 | | | 11520 |
| Объём данных, Тб | 144 | | | 276480 |
| Объём данных, Пб | 0,140625 | | | 270 |
| Потребляемая мощность, Вт | 250 | | | 480000 |

Модуль энергообеспечения представляет собой космический аппарат, оснащённый солнечными батареями большой площади и при необходимости оборудован для преобразования электроэнергии. Согласно предварительным расчётам, для питания компьютерного оборудования одного из модулей хранения и обработки информации, необходима мощность 480 кВт.

Один из основных способов получения электрической энергии на космических аппаратах – солнечные батареи: они работают долгое время без расхода каких-либо материалов, и в то же время являются экологически безопасными, в отличие от ядерных и радиоизотопных источников энергии.

Таблица 2

| Тип фотоэлементов и модулей | Коэффициент фотоэлектрического преобразования, % |
|---|--|
| Кремниевые | |
| Si (кристаллический) | 24,7 |
| Si (поликристаллический) | |
| Si (тонкопленочная передача) | |
| Si (тонкопленочный субмодуль) | 10,4 |
| III-V | |
| GaAs (кристаллический) | 25,1 |
| GaAs (тонкопленочный) | 24,5 |
| GaAs (поликристаллический) | 18,2 |
| InP (кристаллический) | 21,9 |
| Тонкие плёнки халькогенидов | |
| CIGS (фотоэлемент) | 19,9 |
| CIGS (субмодуль) | 16,6 |
| CdTe (фотоэлемент) | 16,5 |
| Аморфный/Нанокристаллический кремний | |
| Si (аморфный) | 9,5 |
| Si (нанокристаллический) | 10,1 |
| Фотохимические | |
| На базе органических красителей | 10,4 |
| На базе органических красителей (субмодуль) | 7,9 |
| Органические | |

| | |
|-------------------------------|------|
| Органический полимер | 5,15 |
| Многослойные | |
| GaInP/GaAs/Ge | 32,0 |
| GaInP/GaAs | 30,3 |
| GaAs/CIS (тонкопленочный) | 25,8 |
| a-Si/mc-Si (тонкий субмодуль) | 11,7 |

Также, в 2018 году, с открытием флексо-фотовольтаического эффекта, обнаружена возможность увеличения КПД фотоэлементов [6], а так же за счёт продления жизни горячих носителей (электронов) теоретический предел их эффективности поднялся с 34 сразу до 66 % [7].

Таким образом, площадь солнечных батарей для одной части серверного комплекса должна быть около 500 м², что является достаточно большой величиной, однако, с учётом недавних разработок в области гелиоэнергетики площадь солнечных панелей может быть сокращена практически вдвое.

Библиографический список

1. Замятин, Д. А. Концепция орбитальной компьютерной сети / Д. А. Замятин, В. С. Богданова, В. В. Кольга // Актуальные проблемы авиации и космонавтики : материалы III Междунар. науч. конф. (10–14 апреля 2017 г., Красноярск) : в 2 ч. / под общ. ред. Ю. Ю. Логинова. – Красноярск, 2017. – URL: <http://elibrary.ru>
2. Замятин, Д. А. Основные принципы структуры орбитальной компьютерной сети / Д. А. Замятин, В. С. Богданова, В. В. Кольга // Решетневские чтения : материалы XXI Междунар. науч.-практ. конф., посвящ. памяти генерального конструктора ракетно-космических систем академика М. Ф. Решетнева (8–11 ноября 2017 г., Красноярск) : в 2 ч. / под общ. ред. Ю. Ю. Логинова. – URL: <https://reshetnev.sibsau.ru/page/materialykonferentsii>
3. Замятин, Д. А. Концепция независимой спутниковой компьютерной сети / Д. А. Замятин, В. С. Богданова, С. В. Степанова // Новые информационные технологии и системы : сб. науч. ст. XIV Междунар. науч.-техн. конф., посвящ. 70-летию кафедры «Вычислительная техника» и 30-летию кафедры «Системы автоматизированного проектирования» (22–24 ноября 2017 г., Пенза). – Пенза : Изд-во ПГУ, 2017. – 420 с.
4. TS-453Bmini | QNAP. – URL: <https://qnap.ru/ts-453bmini> (дата обращения: 14.04.2018).
5. Обзор инженерных систем ЦОД: Модульный ЦОД. – URL: <http://alldc.ru/documentation/document/613.html> (дата обращения: 14.09.2016).
6. Физики выдавили из солнечных батарей дополнительную энергию. – URL: <https://nplus1.ru/news/2018/04/20/flexo-photovoltaic-effect> (дата обращения: 20.07.2018).
7. Химики продлили жизнь горячим электронам в перовскитных батареях. – URL: <https://nplus1.ru/news/2018/01/17/hot-carriers> (дата обращения: 20.07.2018).

К. И. Никишин, Е. И. Гурин, Н. Н. Коннов

Пензенский государственный университет, Пенза, Россия

РЕАЛИЗАЦИЯ SCHEDULE TIME-TRIGGERED ETHERNET С ИСПОЛЬЗОВАНИЕМ ПЛИС

Рассматривается реализация усовершенствованного способа передачи Ethernet сообщений в распределенной системе жесткого реального времени – Schedule Time-Triggered Ethernet (STTE). Кадры передаются согласно заранее установленному расписанию. STTE реализован с использованием языка VHDL в среде Xilinx.

На сегодняшний момент Ethernet является подавляющей сетевой технологией для локальных сетей LAN. Постоянно растут новые требования к работе коммуникационных

протоколов, накладывая тем самым новые ограничения и адаптации к Ethernet для критически важных приложений.

Однако технология Ethernet не позволяет обрабатывать трафик реального времени с минимальной задержкой, для этих целей предусмотрена технология Time-Triggered Ethernet (TTE) [1–3].

Недостатками технологии TTE являются:

- Аппаратные средства на сбор, вычисление, хранение константной задержки в формате ТТ-кадра, а также затраты на подключение пользовательской аппаратуры через специальное устройство «защитник системы».
- Прерывание ЕТ-кадра в случае временного конфликта между ЕТ- и ТТ-кадрами в коммутаторе, что приводит к уменьшению пропускной способности сети из-за повторной передачи прерванного ЕТ-кадра. Появляется дополнительная нагрузка вычислительных средств на передатчик по контролю за временем отправления.
- Отсутствие возможности передачи нескольких кадров, направленных к разным портам, в одном таймслоте.

Авторы предлагают усовершенствованный способ передачи Ethernet сообщений в распределенной системе жесткого реального времени по технологии TTE – Schedule Time-Triggered Ethernet (STTE). Механизм STTE позволяет упростить подключение пользовательской аппаратуры, уменьшить задержку кадров в сети и исключить временные конфликты между Ethernet сообщениями различного класса.

Основная идея алгоритма STTE заключается в реализации функций по обеспечению доставки ТТ-кадров абоненту, согласно заранее установленному расписанию не защитником системы, реализующего селекцию кадров на входе коммутатора, а непосредственно коммутатором Ethernet на его выходном порту.

Рассматриваемы в статьях [4–5] механизм передачи кадров с использованием процессорного ядра MicroBlaze не подходит для детального рассмотрения передачи кадров по STTE, поскольку использование процессорных ядер и высокоуровневой логики программирования может приводить значительной задержке.

Создается сеть из нескольких ПЛИС, поскольку передача кадров от ПК к отладочной плате может приводить к серьезным отклонениям в работе и затрудняет работу отладки. Поэтому для этих целей было принято решение объединить две ПЛИС, таким образом, чтобы одна ПЛИС являлась передатчиком, т.е. генерировала кадры для контроллера Ethernet, а вторая ПЛИС являлась приемником сгенерированных кадров. Как было сказано ранее включение режима передатчика осуществляется с помощью активного сигнала TX_EN. Реальная организация устройства представлена на рис. 1.

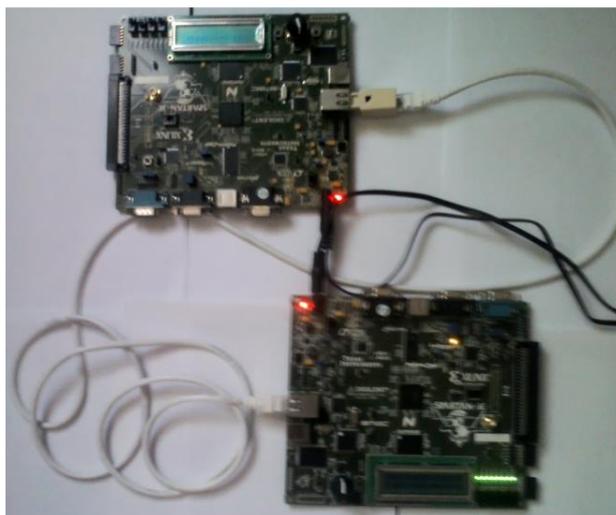


Рис.1. Реальная организация устройства по технологии STTE

Аппаратная организация архитектуры устройства представлена на рис. 2.

Для организации взаимодействия сети ПЛИС было осуществлена программная прошивка ПЛИС в разных режимах. Для передатчика ПЛИС была прошита память ПЗУ платы через программатор iMPACT [6–7], таким образом, при включении отладочной платы начинают передаваться кадры через контроллер Ethernet. Для прошивки памяти необходимо создать файл прошивки памяти с расширением .mcs.

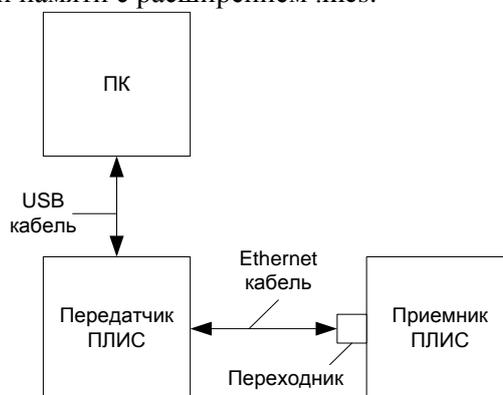


Рис. 2. Аппаратная организация архитектуры устройства

В случае приемника прошивается непосредственно сама ПЛИС через программатор iMPACT, для прошивки ПЛИС используется скомпилированный файл с расширением .bit. Для осуществления корректного обмена кадров между устройствами был подключен переходник к приемнику, для того чтобы образовать перекрестное соединение. Для отладки подключалась плата передатчика, поскольку она генерирует передачу кадров по технологии STTE, и временные диаграммы будут рассматриваться на этой стороне. Приемник лишь принимает полученные кадры. Устройство обрабатывает кадры реального времени (ТТ-кадры стандарт SAE TTE [8]) и кадры стандарта Ethernet (ЕТ-кадры) [9].

Разработанное устройство обрабатывало следующие ситуации:

- передача ТТ-кадра, который пришел во время, то есть время доставки в расписании равно моменту прихода ТТ-кадра;
- ТТ-кадр пришел раньше времени в расписании и осуществляется проверка ЕТ-кадра. Если размер ЕТ-кадра позволяет пройти до времени передачи ТТ-кадра передается ТТ-кадра, а затем ТТ-кадр, когда достигнуто время из расписания;
- в случае, когда размер ЕТ-кадра больше времени доставки ТТ-кадра, тогда передается ТТ-кадр по расписанию, а только за тем ЕТ-кадр;
- прерывание ТТ-кадра, пришедшего позже времени доставки.

В статье рассмотрена реализация усовершенствованного способа передачи Ethernet сообщений в распределенной системе жесткого реального времени - Schedule Time-Triggered Ethernet (STTE). Кадры передаются согласно заранее установленному расписанию. STTE реализовался с использованием языка VHDL в среде Xilinx. Рассмотрена организация устройства в общем виде и реальная структура устройства, а также рассмотрены ситуации обработки кадров разного вида.

Библиографический список

1. Kopetz, H. The time-triggered Ethernet (TTE) design / H. Kopetz, A. Ademaj, P. Grillinger, K. Steinhammer // Proc. of 8th IEEE International Symposium on Object-oriented Real-time Distributed Computing (ISORC). – Washington, 2005. – P. 22–33.
2. Steinhammer, K. A time-triggered Ethernet (TTE) switch / K. Steinhammer, P. Grillinger, A. Ademaj, H. Kopetz // Proc. of the Conference on Design, Automation and Test in Europe (DATE). – Belgium, 2006. – P. 794–799.
3. Никишин, К. И. Моделирование сетями Петри коммуникационного оборудования Time-Triggered Ethernet / К. И. Никишин, Н. Н. Коннов // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 222–236.

4. Никишин, К. И. Аппаратная конфигурация ПЛИС типа FPGA с использованием интерфейса Ethernet для обмена информацией / К. И. Никишин // Новые информационные технологии и системы : сб. науч. ст. XIV Междунар. науч.- техн. конф. – Пенза : Изд-во ПГУ, 2017. – С. 117–121.
5. Никишин, К. И. Обмен информацией с использованием интерфейса Ethernet на основе ПЛИС типа FPGA / К. И. Никишин // Новые информационные технологии и системы : сб. науч. ст. XIV Междунар. науч.- техн. конф. – Пенза : Изд-во ПГУ, 2017. – С. 121–123.
6. Руководство к плате Spartan-3E Start Kit Board User Guide UG230 2006.
7. Описание работы с программными средами фирмы Xilinx. – URL: <http://xilinx.com/>
8. SAE Standard AS 6802: Time-Triggered Ethernet – SAE International, 2011. – P. 108. – URL: <http://docs.cntd.ru>
9. Никишин, К. И. Моделирование систем на базе технологии Time-Triggered Ethernet / К. И. Никишин, Н. Н. Коннов, Д. В. Пащенко // Информационные технологии и математическое моделирование (ИТММ–2016) : материалы XV Междунар. конф. имени А. Ф. Терпугова. – Томск : Изд-во Томского ун-та, 2016. – Ч. 2. – С. 117–122.

И. Ю. Фролов

Пензенский государственный университет, Пенза, Россия

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ СИНХРОННОГО И АСИНХРОННОГО СБРОСА

Рассматриваются некоторые особенности обработки сигнала сброса на входе триггеров с синхронной и асинхронной логикой.

Сигнал «сброс» (англ. *reset*) используется для приведения аппаратуры – как отдельных триггеров, так и более крупных блоков – в известное исходное состояние. Обработка сброса в асинхронном и синхронном режимах имеет свои особенности, которые следует учитывать при разработке с использованием языков описания аппаратуры.

Синхронный сброс

В синхронных схемах все части схемы подчиняются глобальному генератору тактовых импульсов. Соответственно, синхронный сброс может произойти только на активном фронте синхроимпульса.

В языках описания аппаратуры (в примерах используется VHDL) синхронный сброс задается следующим образом: добавляется условие *if...else*, где под условие «*if*» заводится сброс, а в «*else*» – остальная логика сигнала.

```
process( clk )
begin
  if rising_edge( clk ) then
    if reset = '1' then
      <Сброс>
    else
      <Остальная логика>
    end if;
  end if;
end process;
```

Пример 1 – Описание синхронного сброса

Разбирая пример 1, обратим внимание на следующее:
– сигналу сброса уделяется наивысший (помимо синхросигнала) приоритет при описании логики;

– в списке чувствительности процесса указывается только синхрочастота;
 – условие работы по нарастающему фронту синхросигнала (“if rising edge”) делает сброс синхронным.

Следование предложенному выше стилю поможет избежать ситуации, когда некоторые симуляторы при моделировании работы схемы могут заблокировать прохождение сигнала сброса ко входу триггера. Также стоит отметить, что, хотя предложенный стиль и увеличивает количество вентилей перед входом триггера, этим можно пренебречь.

Преимущество синхронного сброса в заметном упрощении работы симулятора. Кроме того, синхрочастота подавляет очень короткие импульсы входного сигнала (главное, чтобы изменения сигнала не были около нарастающего фронта синхрочастоты, иначе может возникнуть метастабильное состояние триггера).

Помимо вероятности метастабильности к недостаткам относится то, что сигнал сброса должен быть активным не менее периода синхросигнала, иначе есть вероятность, что сброс не произойдет.

Асинхронный сброс

Смысл данной реализации в независимости сброса от синхросигнала: когда сигнал сброса появился на входе триггера, тогда и будет обработан. Асинхронный сброс широко применяется при разработке аппаратуры.

Для корректного функционирования аппаратуры предлагается использовать следующий стиль:

```
process( clk, reset )
begin
  if reset = '1' then
    <Сброс>
  elsif rising_edge( clk ) then
    <Остальная логика>
  end if;
end process;
```

Пример 2 – Описание асинхронного сброса

В данном случае, в отличие от примера 1, все наоборот: сигнал сброса независим от синхрочастоты и добавлен в список чувствительности (стоит отметить, что в список чувствительности стоит добавлять те сигналы, которые непосредственно влияют на выход триггера).

Основное преимущество подхода в том, что, пока в библиотеке примитивов существуют триггеры с асинхронным сбросом, данные не будут связаны логикой с сигналом сброса. Другими словами, при описании асинхронного сброса в тракт данных не будет добавлена дополнительная логика обработки сброса.

Среди недостатков использования асинхронного сброса можно выделить проблему асинхронности снятия сигнала сброса (когда снятие происходит около активного фронта синхросигнала и может перевести триггер в метастабильное состояние). Кроме того, при выдаче сброса необходимо гарантировать, что сигнал будет активен не менее периода синхрочастоты (как и в случае асинхронного сброса). Так как сброс будет выдаваться из одной точки ко многим триггерам, он будет иметь большое разветвление; поэтому под сброс в кристалле обычно задействуется одна из глобальных цепей. После размещения проекта на кристалле необходимо провести временной анализ этой цепи на предмет соответствия требованиям синхронизации. Из других недостатков можно отметить возможность появления случайных помех, которые будут восприняты как активный сигнал; а так же дополнительные сложности при выполнении статического временного анализа.

Итог. Существует расхожее мнение, что синхронный сброс строго лучше асинхронного. Однако это не совсем так. Плюсы и минусы есть в обоих исполнениях. Например, в некоторых схемах сигнал сброса может формироваться внутренними условиями. В таких случаях, действительно, лучше применять синхронный сброс. И, наоборот, если в проекте, например, используется технология “Clock gating” (отключение синхрочастоты в неиспользуемых цепях), то синхронный сброс будет неэффективен.

Библиографический список

1. Бибило, П. Н. Основы языка VHDL / П. Н. Бибило. – М. : Либроком, 2016. – 328 с.
2. Несколько слов о «сбросе»: описание. – URL: http://www.kit-e.ru/articles/circuit/2009_1_96.php (дата обращения: 09.07.2018).
3. Cummings, C. E. Asynchronous & Synchronous Reset / C. E. Cummings, D. Mills, S. Golson. – Boston, 2003. – 36 с.

К. Ю. Шабалова

Пензенский государственный университет, Пенза, Россия

ПРОБЛЕМА ОРИЕНТАЦИИ МОБИЛЬНЫХ УСТРОЙСТВ В UNITY 3D

Рассматриваются основные проблемы ориентации мобильных устройств в Unity 3D, а также их варианты решений.

Unity 3D предоставляет разработчикам широкий ассортимент платформ для разработки приложений, среди которых так же числятся и различные мобильные платформы.

Одна из ключевых особенностей разработки UI для мобильных устройств – большая вариативность соотношений сторон экрана. Если сравнивать это число с разработкой под ПК или консоли, то для этих платформ применяются, либо альбомная ориентация (когда ширина больше высоты), либо, значительно реже, квадратная. У телефонов же добавляется еще одна, существенно отличающаяся от двух предыдущих – портретная [1].

Портретная ориентация являет собой соотношение сторон, обратное пропорциональное альбомной ориентации, вследствие чего при плотной компоновке UI элементов [3] могут возникать проблемы размещения всех элементов на экране во время смены ориентации с альбомной на портретную.

Далее будут рассмотрены несколько самых популярных в среде Unity 3D способов решения данной проблемы.

Первый способ – изменение масштабов интерфейса в соответствии с размерами ширины или длины экрана, путем выставления значения `scalewithscreenize` параметру `UIScaleMode` компонента `CanvasScaler`. Данный способ прост в применении и не требует большого количества времени для внесения изменений в дальнейший функционал для его корректной работоспособности.

Однако следует учитывать, что размеры большинства телефонов невелики и при изначальной верстке под альбомную ориентацию портретной ориентации кнопки могут оказаться чересчур маленькими и, вследствие чего, неудобными, и, так же наоборот: при верстке под портретную – в альбомной они могут оказаться громоздкими. Ввиду этих обстоятельств на разработку интерфейса накладывается ряд ограничений, по причине которых этот способ не является универсальным и может подходить только в конкретных ситуациях.

Второй способ – блокировка смены ориентации. Данный способ легко применим в тех случаях, когда разрабатываемое приложение в каждой сцене должно находиться в одной и той же ориентации, и в этом случае в разделе `ResolutionandPresentation` раздела `PlayerSettings` можно выставить доступные одно или несколько ориентаций. Однако при необходимости в разных сценах включать различные ориентации или отключать блокировку – придется настраивать интерфейс вручную, через код.

Есть несколько нюансов, связанных с настройкой ориентации устройства через код. Первый – в UnityEngine (пространство имен функционала Unity) нет специального события [4], которое бы извещало подписанные на него методы о смене ориентации устройства. Поэтому программисту необходимо самому проверять, какое состояние сейчас активно, как правило, в теле метода `Update` [2]. Второй нюанс заключается в том, что у большинства устройств помимо вариаций альбомной и портретной ориентации есть еще два состояния: лицом вверх и лицом вниз. Эти состояния являются общими как для альбомной, так и для

портретной ориентации и не несут в себе информации о том, какое из состояний (альбомное или портретное) должно быть активно, что в некоторых случаях может приводить к неверной отрисовке интерфейса. Доработка этого функционала так же ложится на программиста.

Третий способ – верстка отдельного интерфейса, как для альбомной, так и для портретной ориентации. В Unity нет встроенного функционала для данного способа, и его необходимо реализовывать программно со всеми вышеуказанными проблемами настройки ориентации устройства через код. Данный способ самый время затратный, поскольку требуется время на создание дополнительного интерфейса, а так же все последующие правки по UI будут предполагать их внесение в двойном объеме.

Заключение

В данной статье рассматривались решения проблем ориентации мобильных устройств в Unity 3D. Данные решения применяются при плотной компоновке UI на экране мобильного устройства.

Библиографический список

1. UnityDocumentation: Ориентация экрана. – URL: <https://docs.unity3d.com/ru/current/Manual/AndroidMobileAdvanced.html> (дата обращения: 21.07.2018).
2. Android: Ориентация. – URL: <http://developer.alexanderklimov.ru/android/orientation.php> (дата обращения: 21.07.2018).
3. UnityUI в версии 4.6: Система якорей. – URL: <https://habr.com/post/248273/> (дата обращения: 21.07.2018).
4. Events: события в Unity 3D. – URL: <https://unity3d.com/ru/learn/tutorials/topics/scripting/events> (дата обращения: 22.07.2018).

А. А. Турмышов, Н. Ю. Митрохина

Пензенский государственный университет, Пенза, Россия

ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ЗАДАЧ В КУРСЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА»

Приведены примеры двух задач по теоретической механике из раздела «Динамика», которые могут быть решены с применением информационных систем компьютерного моделирования. Для каждой задачи приведена математическая модель, а также результаты моделирования механического процесса или явления.

В настоящее время существует большое количество информационных систем проектирования и моделирования, позволяющих применять их при решении различных научно-технических задач. Такие широко известные математические пакеты как *Mathcad*, *Maple*, *Matlab* и др. среды содержат в себе практически все известные методы как аналитического, так и численного решения: операции с векторами и матрицами, дифференцирование и интегрирование, решение дифференциальных уравнений, построение двумерных и трехмерных графиков, программирование на языке пакета, статистический анализ и многое другое.

Рассмотрим возможность применения математического пакета *Mathcad* для компьютерного моделирования некоторых задач из раздела «Динамика» курса «Теоретическая механика». Данная интегрированная программная система автоматизации математических расчетов расширяет границы ее использования для решения задач различного научно-технического характера без применения кодирования на каком-либо алгоритмическом языке. Это позволяет применять *Mathcad* при изучении многих тем курса «Теоретическая механика» и исследовательской деятельности студентов различных технических направлений, начиная с первого курса обучения.

Задача 1. Моделирование движения тела, брошенного под углом к горизонту.

Постановка задачи. Тело массой m , брошено под углом α к горизонту с начальной скоростью v_0 . Принимая тело за материальную точку, необходимо воспроизвести и исследовать его движение.

Тело движется в реальной среде под действием силы тяжести $P = mg$ и силы сопротивления среды R , пропорциональной скорости движения (рис. 1,а). Построение математической модели заключается в составлении дифференциальных уравнений движения в проекциях на декартовы оси:

$$\begin{aligned} m \frac{d^2 x}{dt^2} &= -R \cdot \cos \alpha, \\ m \frac{d^2 y}{dt^2} &= -P - R \cdot \sin \alpha. \end{aligned} \quad (1)$$

Для численного решения задачи удобнее всего представить систему дифференциальных уравнений второго порядка в виде системы уравнений первого порядка. Преобразовав данные уравнения (1), получим:

$$\begin{aligned} \frac{dv_x}{dt} &= -\mu v_x, & \frac{dx}{dt} &= v_x, \\ \frac{dv_y}{dt} &= -g - \mu v_y, & \frac{dy}{dt} &= v_y, \end{aligned}$$

где μ – коэффициент пропорциональности.

Численное решение дифференциальных уравнений в *Mathcad* можно получить, используя разностные уравнения, собственные процедуры (программирование), а также встроенные функции *odesolve*, *rkfixed*, *rkadapt* и другие [2]. При использовании функции *odesolve* сначала необходимо ввести постоянные параметры уравнения, а затем после слова *Given* записать дифференциальные уравнения и начальные условия в привычном виде:

$$g := 9.8 \quad t := 0, 0.01..40 \quad \mu := 2 \quad \alpha := \frac{\pi}{4} \quad x_0 := 0 \quad y_0 := 0 \quad v_0 := 20$$

Given

$$\frac{d}{dt} v_x(t) = -\mu \cdot v_x(t) \quad v_x(0) = v_0 \cdot \cos(\alpha) \quad \frac{d}{dt} v_y(t) = -g - \mu \cdot v_y(t) \quad v_y(0) = v_0 \cdot \sin(\alpha)$$

$$\frac{d}{dt} x(t) = v_x(t) \quad x(0) = x_0$$

$$\frac{d}{dt} y(t) = v_y(t) \quad y(0) = y_0$$

$$\begin{pmatrix} v_x \\ v_y \\ x \\ y \end{pmatrix} := \text{Odesolve} \left[\begin{pmatrix} v_x \\ v_y \\ x \\ y \end{pmatrix}, t, 40 \right]$$

Данная программа, реализованная в среде *Mathcad*, позволяет провести вычислительный эксперимент по исследованию траектории тела, брошенного под углом горизонту, в зависимости от начальной скорости, угла полета и коэффициента сопротивления среды. Полученные результаты моделирования движения тела (рис. 1, б) также могут быть использованы при моделировании задачи внешней баллистики [1] в рамках изучения профессиональных дисциплин некоторых специальностей, а также при самостоятельной научно-исследовательской деятельности студентов.

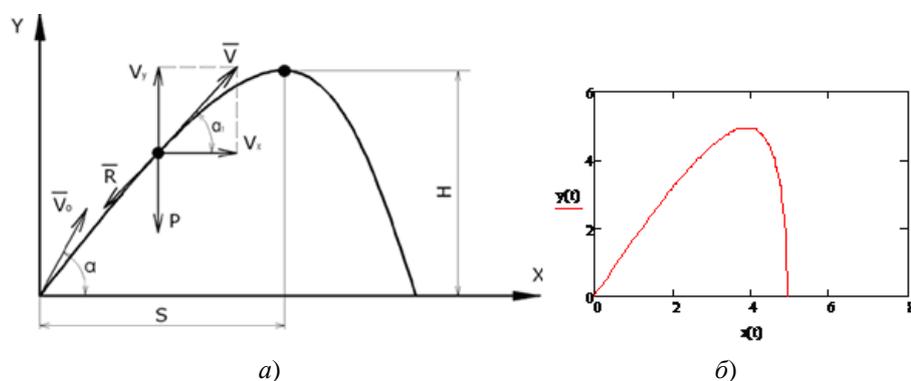


Рис. 1. График траектории (а) и результаты моделирования (б) движения тела, брошенного под углом к горизонту

Задача 2. Моделирование прямолинейных колебаний материальной точки

Постановка задачи. Дана механическая система, состоящая из груза заданной массы m , закрепленной с помощью системы пружин заданной жесткости c_1, c_2, c_3 и демпфера с заданным коэффициентом сопротивления μ . На груз действует возмущающая сила, изменяющаяся по гармоническому закону $Q = Q_0 \sin(pt)$. Начальные условия движения груза: $t_0 = 0, x_0, \dot{x}_0$. Считая груз материальной точкой, исследовать свободные, затухающие и вынужденные колебания.

Данные колебания изучаются в курсе теоретической механики [3], поэтому не будем приводить подробное математическое описание, а сразу приведем дифференциальные уравнения для различных вариантов колебаний:

а) свободные колебания – $\ddot{x} + k^2x = 0$, где k – циклическая (круговая) частота свободных колебаний;

б) затухающие колебания – $\ddot{x} + 2n\dot{x} + k^2x = 0$, где n – коэффициент затухания колебаний;

в) вынужденные колебания без учета сопротивления среды – $\ddot{x} + k^2x = h\sin(pt)$, где $h = \frac{Q_0}{m}$;

г) вынужденные колебания с учетом сопротивления среды – $\ddot{x} + 2n\dot{x} + k^2x = h\sin(pt)$.

Если частота собственных колебаний k и частота вынуждающего воздействия p равны ($k = p$), то возникает такое широко известное явление как резонанс. Условие близости частот собственных колебаний и вынуждающего воздействия является условием возникновения явления биений.

Таким образом, в зависимости от действующих сил математической моделью, описываемых процессов, будут являться дифференциальные уравнения с заданными начальными параметрами.

Проведенный вычислительный эксперимент в среде *Mathcad* позволяет исследовать прямолинейные колебания материальной точки под действием различных сил в зависимости от начальных условий задачи (рис. 2, а-е), что дает студентам наглядное представление об изучаемых процессах.

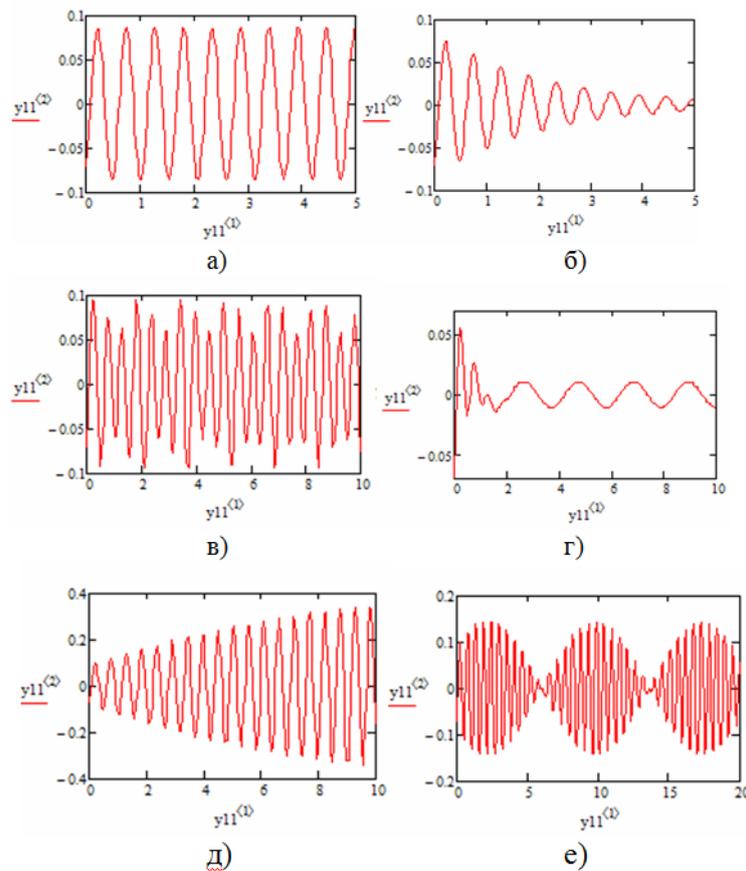


Рис. 2. Моделирование колебательного процесса

а – свободные колебания; б – затухающие колебания; в – вынужденные колебания без учета сопротивления среды; г – вынужденные колебания с учетом сопротивления среды; д – явление резонанса; е – явление биений

В результате использования информационных систем в преподавании курса «Теоретическая механика» на примерах компьютерного моделирования ряда задач и наглядной

демонстрации физических процессов и явлений достигается повышение мотивации к обучению и активизация познавательной деятельности у студентов. Это позволяет обучающимся расширять возможности для самостоятельной научно-исследовательской и творческой деятельности, а также закреплять различные профессиональные навыки.

Библиографический список

1. Дмитриевский, А. А. Внешняя баллистика : учебник для студентов вузов / А. А. Дмитриевский, Л. Н. Лысенко. – 4-е изд., перераб. и доп. – М. : Машиностроение, 2005. – 608 с.
2. Маликов, Р. Ф. Практикум по компьютерному моделированию физических явлений и объектов : учеб. пособие / Р. Ф. Маликов. – Уфа : Изд-во БашГПУ, 2005. – 291 с.
3. Яблонский, А. А. Курс теоретической механики / А. А. Яблонский. – М. : Высш. шк., 1977. – Ч. 2. Динамика. – 430 с.

А. П. Писарев, Е. М. Голобокова, Ю. В. Терякова

Пензенский государственный университет, Пенза, Россия

НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Рассмотрена актуальность использования быстро развивающихся направлений новых информационных технологий в учебном процессе, что позволит реализовать на новом качественно более высоком уровне классический принцип дидактики – принцип наглядности.

В настоящее время значительно увеличилась роль информационных технологий в жизни людей. Современное общество включилось в общеисторический процесс, называемый информатизацией. Одним из приоритетных направлений процесса информатизации современного общества является информатизация образования. Современные тенденции по увеличению роли и внедрению информационных технологий в систему образования обеспечивают процессы информатизации современного общества и тесно связанные с ними процессы информатизации всех форм образовательной деятельности, которые в свою очередь характеризуются процессами совершенствования и массового распространения современных информационных и коммуникационных технологий (ИКТ).

Внедрение персонального компьютера в информационную сферу и применение современных телекоммуникационных средств связи определили новый этап развития информационной технологии.

Целью новых информационных технологий в современном образовании являются (рис. 1).

Новые информационные технологии базируются на следующих основных принципах:

- интерактивный (диалоговый) режим работы с компьютером;
- интегрированность и взаимодействие с другими программными продуктами и средами;

- гибкость процесса изменения данных и постановок задач [1].

Соблюдение этих принципов позволит решать следующие задачи:

- упрощение получения и осознания необходимой информации, что влечет повышение уровня знаний;

- систематизация информации с использованием электронных библиотек, справочников и каталогов;

- отработка различных навыков и умений, проведение удаленных научных и лабораторных экспериментов;

- моделирование взаимодействия объектов с целью их изучения;

- обмен информацией между пользователями, находящимися на большом расстоянии друг от друга;
- визуализация информации и ее демонстрация с использованием мультимедийных технологий (например, на презентациях и web-страницах).



Рис. 1. Цели новых информационных технологий в образовании

Когда сегодня говорят об информационных технологиях в образовании, зачастую подразумевают мультимедийные технологии, которые помогают более глубоко исследовать многие вопросы, и самое главное, сокращают время на изучение материала.

Мультимедиа представляет собой текстовую, графическую, видео-анимационную и звуковую информацию представленную в цифровом виде, а также возможность интерактивно взаимодействовать с ней. Т.е. мультимедиа позволяют одновременно работать с видео изображением, графикой, текстом и звуком, и при этом пользователю, как правило, отводится активная роль.

Мультимедийные технологии сегодня – одно из наиболее быстро развивающихся направлений новых информационных технологий, используемых в учебном процессе и ее важная особенность – интерактивность.

Специалист в области искусственного интеллекта Д. А. Пospelов определил три основные задачи когнитивной (компьютерная графика для научных абстракций, способствующая рождению нового научного знания) компьютерной графики:

- создание моделей представления знаний, в которых была бы возможность однообразными средствами представлять, как объекты, так и образы-картины;
- визуализация тех человеческих знаний, для которых пока невозможно подобрать текстовые описания;
- поиск путей перехода от наблюдаемых образов-картин к формулировке гипотез, которые скрыты за динамикой наблюдаемых картин [2].

Основой образовательного процесса при очном обучении являются лекции, формой, адекватной уровню развития информационных технологий, следует признать мультимедийные курсы лекций, читаемые в специально оборудованных учебных аудиториях. Разработанные преподавателем мультимедийные курсы могут применяться как для индивидуального дистанционного, так и для группового обучения с интерактивными свойствами контроля усваиваемых знаний. Современные мультимедийные технологии позволяют соединить информацию текстового, графического, анимационного характера для демонстрации материалов изучаемых курсов. Это дает возможность реализовать на новом качественном более высоком уровне классический принцип дидактики – принцип наглядности [3].

В настоящее время наука подошла к такому уровню своего развития, что появилась возможность создания искусственного интеллекта.

Искусственный интеллект (ИИ) – это способ сделать компьютер, компьютерконтролируемого робота или программу способную также разумно мыслить, как человек [4]. Искусственный интеллект – наука и технология, основанная на таких дисциплинах, как информатика, биология, психология, лингвистика, математика, машиностроение. Одним из главных направлений искусственного интеллекта – разработка компьютерных функций, связанных с человеческим интеллектом, таких как: рассуждение, обучение и решение проблем.

Польза и перспектива развития искусственного интеллекта в образовании:

ИИ позволит подбирать форму обучения для каждого человека индивидуально, исходя из его способностей, затрат времени на полное понимание и освоение учебного материала.

ИИ будет полезен в быстрой, правильной и справедливой проверке знаний после обучения, что гораздо упростит и ускорит оценивание.

Человек сможет заниматься самообразованием при помощи ИИ, а не путем получения знаний от другого человека. Роботы могут не только обучать автономно от учителя, но и помогать ему.

В образовательном секторе происходит новый виток развития. И связано оно с использованием ботов – программ на основе искусственного интеллекта для самостоятельного общения со студентами. Chatbots – это системы искусственного интеллекта, с которыми пользователи взаимодействуют через текст. Моделируется взаимодействие с человеком, сходное с опытом использования Apple Siri или Amazon's Alexa.

При обилии действующих онлайн-сервисов в сегменте eLearning, чат-боты представляются многообещающим инструментом, поскольку могут сопровождать каждого слушателя индивидуально, в соответствии с его уровнем и выбранным темпом освоения материала, делая обучение доступным практически для любого человека, имеющего доступ к Wi-Fi. По сравнению с традиционным тьюторингом, чат-боты не требуют существенных ресурсных затрат, и потенциально могут помочь миллионам студентов по всему миру.

Одной из областей образования, где чат-боты могут вскоре занять серьёзную нишу, является изучение языка. В качестве примера достаточно привести популярное приложение Duolingo, создатели которого одними из первых стали использовать ботов. Для Duolingo были разработаны несколько персонажей для «живых» разговоров на иностранном языке, чтобы студенты могли практиковать свой словарный запас и проверять правильность грамматических конструкций.

По сравнению с традиционным программным обеспечением для обучения языку, чат-боты гораздо более гибкий и адаптируемый инструмент. Они реагируют по-разному в зависимости от ответа, который дает пользователь, учитывая различные варианты, как в реальных диалогах.

Роботы не смогут заменить человека в аудитории в ближайшее время, но уже могут взять часть его рабочей нагрузки на себя.

Предметные тесты и эссе отнимают у преподавателя значительное время на проверку, и при этом задания в них часто повторяются. Особенно ярко это проявляется в массовых открытых онлайн-курсах (МООС), где счёт идёт на сотни и тысячи студентов, что делает индивидуальную обратную связь практически недоступной роскошью.

Возможность машинного обучения делает Chatbots столь многообещающей технологией в области образования. Их база знаний, при активном использовании, будет только расти, а сами боты смогут потенциально учиться вместе со студентами уже через несколько лет.

Внедрение новых технологий влечет за собой переформатирование всего учебного процесса, с целью адаптации к использованию новых возможностей изучения теории и отработки усвоенных знаний на практике [5].

Технологии виртуальной реальности предоставляют отличные возможности для того, чтобы усвоить материал эмпирического характера. Традиционный формат урока практически не меняется, а лишь дополняется погружением в VR на 5-10 минут. Возможно деление одного занятия на несколько этапов, в каждом из которых наиболее сложные моменты визуализируются в виртуальном мире. Как и раньше, основой изложения нового материала

остается лекция. Но виртуальная реальность дает возможность усовершенствовать урок, вовлекая учеников полностью погрузиться в учебный процесс, визуализируя ключевые моменты пройденного материала.

Виртуальная реальность позволяет избавиться от границ, что могут возникать во время видеоконференций или дистанционных уроков, создавая эффект личного присутствия.

Если существуют обстоятельства, которые мешают посещать занятия, в ученика есть возможность проходить уроки дистанционно. Чтобы это стало возможным, класс или аудиторию необходимо оборудовать специальными камерами, которые позволяют производить съемку в формате кругового обзора (360°) с которых будет транслироваться урок в режиме online. Ученики, которые по той или иной причине не могут присутствовать в классе, могут быть вместе со своими одноклассниками во время урока, конспектируя материал или решая задачи прямо со своего места за партой.

Виртуальная реальность открывает новые возможности для изучения теории и отработки практики, ведь традиционные методы могут быть весьма затратными или слишком сложными. Существует 5 основных преимуществ использования AR/VR в сфере образования.

- **Наглядность.** 3D-графика позволяет воспроизвести детализацию даже самых сложных процессов, невидимых человеческому оку, вплоть до распада ядра атома или химических реакций. Virtual Reality позволяет воспроизвести или смоделировать любые процессы или явления, о которых знает современная наука.

- **Безопасность.** Практические основы управления летательными или сверхскоростными аппаратами, можно абсолютно безопасно отработать на устройстве виртуальной реальности. Еще VR дает возможность отрабатывать сверхсложные медицинские операции или манипуляции, без вреда и опасности для кого-либо.

- **Вовлечение.** VR-технологии дают возможность смоделировать любую механику действий или поведение объекта, решать сложные математические задания в форме игры и прочее. Виртуальная реальность позволяет путешествовать во времени, просматривая основные сценарии важных исторических событий или увидеть человека изнутри на уровне движения эритроцита в крови.

- **Фокусировка.** Пространство, смоделированное в VR можно легко рассмотреть в панорамном диапазоне 360°, не отвлекаясь на внешние факторы.

Рассмотрим основные программные средства учебного назначения, которые используются на практике:

- обучающие программы – это учебные пособия, которые используются слушателями при самостоятельном изучении материала. Обучающие программы позволяют повысить качество усвояемого материала. Обучающие программы, должны быть разработаны на основе учебной программы и отвечать всем требованиям к их содержанию;

- электронный учебник – это обучающая система, которая основана на дидактических и методических материалах по предмету. Электронный учебник используется для самостоятельного изучения материала и как основа лекционного материала с темами, которые предусмотрены учебным планом. Основным преимуществом электронного учебника является графическое представление материала и наличие гиперссылок на материалы других информационных ресурсов. Использование звуковых файлов, анимации, рисунков, видеороликов позволяют повысить уровень усвоения материала.

- интернет – всемирная система объединённых компьютерных сетей для поиска, хранения и передачи информации. Всемирная паутина дает большие возможности для развития образования: дистанционное обучение, доступ к большому количеству информации. Дистанционное обучение – работа преподавателя со студентами на расстоянии, которое отображает все компоненты учебного процесса, что позволяет получить студентам все необходимые знания дистанционно.

Применение на практике в системе образования информационных технологий заставляет по-иному взглянуть на образовательный процесс. Таким образом систему новых ин-

формационных технологий необходимо рассматривать как процесс улучшения качества образования на всех этапах.

Библиографический список

1. Советов, Б. Я. Информационные технологии: теоретические основы : учеб. пособие / Б. Я. Советов, В. В. Цехановский. – СПб. : Лань, 2017. – 444 с. – URL: <https://e.lanbook.com/book/93007>
2. Поспелов, Д. А. Десять «горячих точек» в исследованиях по искусственному интеллекту / Д. А. Поспелов // Интеллектуальные системы (МГУ). – 1996. – Вып. 1–4. – Т. 1. – С. 47–56.
3. Байтуганова, А. О. Мультимедиа технологии в образовании / А. О. Байтуганова, М. Т. Аймбетова, Л. Каужан // Молодой ученый. – 2016. – № 19.2. – С. 9–11. – URL: <https://moluch.ru/archive/123/34439/> (дата обращения: 27.09.2018).
4. Басараб, М. А. Интеллектуальные технологии на основе искусственных нейронных сетей : учеб. пособие / М. А. Басараб, Н. С. Коннова. – М. : МГТУ им. Н. Э. Баумана, 2017. – 56 с.
5. Лепницкий, Д. Технология виртуальной реальности в образовании / Д. Лепницкий. – URL: <https://avblab.com/tehnologii-virtualnoj-realnosti-v-obrazovanii/> (дата обращения: 20.09.2018).

Т. В. Глотова, П. В. Ечная

Пензенский государственный университет, Пенза, Россия

БАЗА ДАННЫХ ПРОГРАММНЫХ СРЕДСТВ, ИСПОЛЬЗУЕМЫХ В ОБУЧЕНИИ

Активизация интереса обучающихся к использованию современных гаджетов и программных сервисов, появление большого числа программных средств, которые можно эффективно использовать в процессе обучения, делают необходимой систематизацию информации об этих средствах и методах обучения и разработки базы данных с целью эффективной организации образовательного процесса.

С развитием и распространением информационных технологий появляется большое количество программных средств и сервисов во всех сферах деятельности человека. В том числе категория образования пополняется новыми разработками, которые можно было бы эффективно применять в процессе обучения [2]. Применение программных средств в процессе обучения способствуют наиболее эффективному восприятию информации учащимся за счет использования различных способов представления материала, прежде всего визуализации нового материала, и оперативного контроля, а преподавателю позволяет получить сведения о степени и качестве усвоения информации. Использование обучающих программных средств ведет к активизации учебного процесса и повышения интереса учащихся к предмету, а так же обеспечивает интерактивность обучения. Визуализация новой информации помогает успешно усваивать новый материал, позволяет учитывать психологические и биологические особенности студентов.

Введение ФГОС ВО с обязательным использованием электронной информационной образовательной среды в российских университетах приводит к интеграции традиционного и дистанционного обучения – смешанному обучению (blending learning). Наличие у обучающихся мобильных устройств, интерес к их активному использованию являются предрасполагающими факторами к применению интерактивных средств обучения, которые могли бы повысить качество обучения. Но поиск таких средств, альтернатив, новинок потребует затратить от преподавателя большое количество времени и усилий. Поэтому хорошим помощником в этом вопросе стала бы информационная система с базой данных обучающих программных средств, с помощью которой можно подобрать необходимое средство по параметрам и помочь в разработке обучающего курса. Использование такой системы суще-

ственно бы облегчило поиск средств обучения для молодых специалистов, а для опытных стало бы инструментом для усовершенствования существующих учебных курсов.

На рис. 1 представлена разработанная модель базы данных, выполненная с использованием CASE-средства DBDesigner.

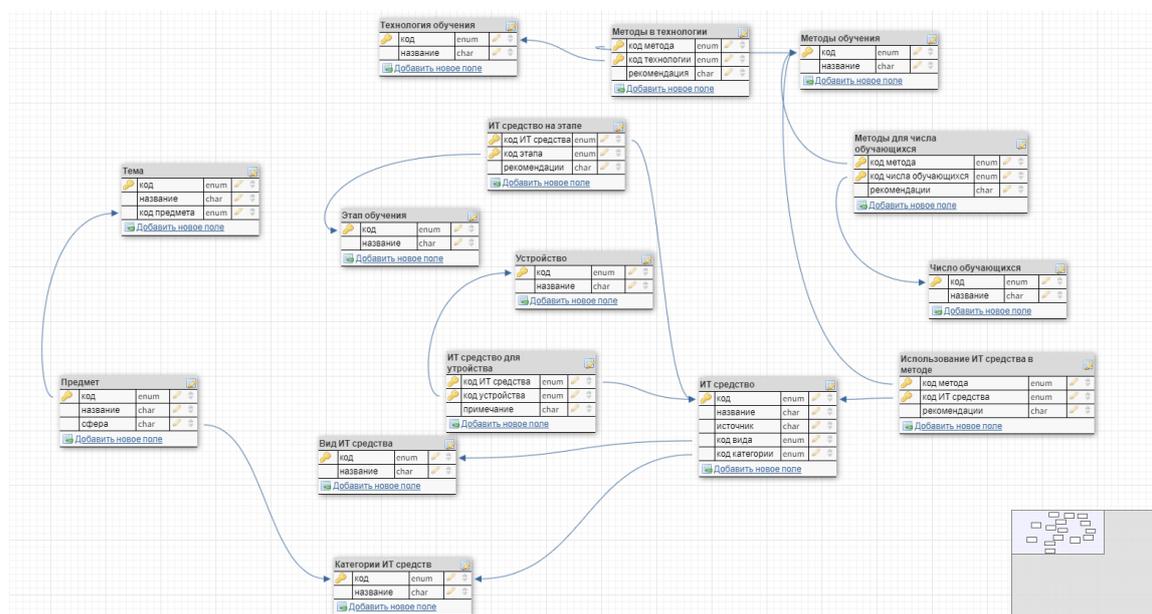


Рис. 1. Модель базы данных

База данных разрабатывается для информационной поддержки преподавателя в подготовке к занятиям: лекциям, лабораторным, практическим работам и др. Разработанная инфологическая модель базы данных содержит в себе следующие сущности: «Предмет», «Тема», «ИТ средство», «Категория ИТ средств», «Вид ИТ средства», «Этап обучения», «Технология обучения», «Устройство», «Методы обучения», «Число обучающихся». Взаимосвязи между всеми сущностями представлены на рис. 1. Предметная область описывает ИТ средства, которые можно использовать в обучении. Указываются возможные для использования этапы обучения, например представление материала, или контроль знаний или умений. Описываются технологии и методы обучения, в том числе активные и интерактивные и их взаимосвязь с ИТ-средствами и работой индивидуально, в парах, в группах, целым классом. Хранятся данные об опыте использования в разработке учебных курсов. Проектирование логической модели базы данных выполнено с использованием метода нормальных форм и соответствует третьей нормальной форме отношений реляционной модели [1].

Разработанная диаграмма вариантов использования (прецедентов) информационной системы, изображена на рис. 2. Диаграмма выполнена по стандарту языка UML [3] в среде CASE-средства StarUML. Пользователем системы является преподаватель, учитель или методист университета, колледжа или школы.

В целях получения рекомендаций или справочной информации пользователь может:

1. Просмотреть методы обучения, технологии обучения, выбрать ИТ средства для обучения на определенном этапе изучения материала, узнать темы для изучения предмета.
2. Получить отчет и краткую справку по использованию ИТ средства и методов обучения.
3. Получить рекомендации по выбору ИТ средства для изучения темы по выбранному предмету.
4. Построить отчет по использованию методов и технологий обучения, используемых ИТ-средств на различных этапах обучения, применительно к преподаваемой дисциплине по различным темам.

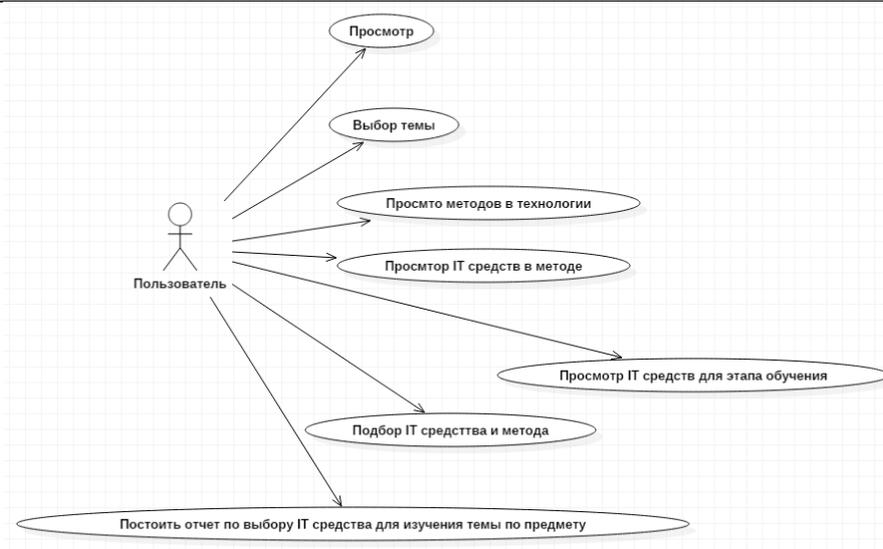


Рис. 2. Диаграмма прецедентов

Планируется применение данной разработки для любой дисциплины и на любом этапе обучения. В систему заложена обобщенная структура обучения, включающая в себя этапы: входное тестирование, теоретическая часть как представление материала, практическая часть для получения практических умений и навыков, итоговое тестирование, но при этом возможна настройка на свои особенности и специфику курса.

Библиографический список

1. Избачков, Ю. С. Информационные системы : учеб. пособие для вузов / Ю. С. Избачков, В. Н. Петров, А. А. Васильев, И. С. Телина. – 3-е изд. – СПб. : Питер, 2011. – 539 с.
2. Ковардакова, М. А. Интерактивные технологии обучения в высшей школе: смешанное обучение : учеб. пособие для слушателей факультета повышения квалификации преподавателей / М. А. Ковардакова. – Ульяновск : УлГУ, 2017. – Ч. 2. – 50 с.
3. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. – М. : ДМК Пресс, 2008. – 496 с.

Т. В. Глотова, О. В. Красильникова

Пензенский государственный университет, Пенза, Россия

КОМПОНЕНТЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Увеличение объема самостоятельной работы студентов в освоении образовательных программ высшего образования требует особого внимания к эффективной организации образовательного процесса с использованием электронного обучения и открытых образовательных ресурсов. Предлагается разработка компонентов информационной системы для поддержки самостоятельной работы студентов, содержащей базу данных об образовательных ресурсах с оценками и комментариями студентов и преподавателей.

Целью разработки компонентов информационной системы является поддержка учебной деятельности студентов с акцентом на самостоятельную работу студентов. В образовательных программах, разработанных в соответствии с последними федеральными образо-

вательными стандартами, большая часть трудоемкости освоения программы отводится на самостоятельную работу студентов. Увеличение объема самостоятельной работы студентов требует такой организации образовательного процесса, чтобы способствовать развитию у студента способностей к самообучению, непрерывному поиску новых, актуальных данных и знаний, переработке и творческому применению знаний.

В настоящее время электронные образовательные ресурсы становятся реально действующим компонентом развития самообразования. Развитие Открытых образовательных ресурсов (Open Educational Resources) – обучающих, учебных или научных ресурсов, размещенных в свободном доступе, либо выпущенных под лицензией, разрешающей их свободное использование или переработку, ведет к увеличению доступности образования, в том числе высшего. Интеграция традиционного и электронного обучения привела к смешанной модели, которая позволяет преподавателю уделять большее время и внимание более интересным и сложным вопросам учебного курса, оставляя многие вопросы на самостоятельную проработку, возможно с поддержкой электронной платформы (использование открытых образовательных ресурсов, компьютерное тестирование, коллективная работа в форуме или чате) [1, 2]. В сети Интернет работает множество различных образовательных платформ и образовательных порталов, размещены видео-уроки, презентации и текстовые материалы по многим образовательным дисциплинам. Но поиск образовательного ресурса по определенной теме и предмету и соответствующий имеющемуся уровню знаний конкретного студента является непростой задачей. Часто такая информация рекомендуется преподавателем дисциплины или передается от студента студенту.

В основном, студенты общаются в социальных сетях, организуя группы по факультетам, учебным группам или конкретным вопросам. Например, в социальной сети «ВКонтакте» существуют группы по вопросам обучения программированию на языке С. Основная идея проекта возникла из необходимости такой организации интерактивного взаимодействия, которое обеспечивает не только общение, но и осуществляет систематизированное хранение данных о просмотренных электронных ресурсах в базе данных и их поиск в дальнейшем студентами университета.

С 2016 года в Пензенском государственном университете действует Электронная информационная образовательная среда (ЭИОС) [3], в которой студент имеет следующие возможности:

- просматривать программу курса и материал, предоставленный преподавателем;
- отправлять файлы на проверку преподавателю, например, курсовые, лабораторные, контрольные работы, рефераты;
- проходить тестирование по вопросам изучаемого предмета;
- наполнять свое портфолио;
- просматривать расписание и результаты обучения;
- общаться с преподавателем и однокурсниками в форуме, организованном преподавателем предмета.

Размещение же материалов самими студентами невозможно, причем в основном представленная информация хранится в текстовых файлах pdf, doc, ppt, поэтому поиск ответа на конкретный вопрос студента среди материалов портала ЭИОС может быть затруднен. Доступ и общение возможно только в разделе изучаемой дисциплины в соответствии с учебным планом. А хотелось бы накапливать информацию и хранить ее в структурированном виде для дальнейшего поиска студентами других курсов и направлений. Студенческий Совет ФВТ ПГУ рассматривает вопрос об организации помощи студентам в вопросах обучения, но на сайте на сегодняшний день отсутствуют данные о такой помощи. Поэтому возникла идея разработки компонентов помощи студентам.

Информационная система поддержки учебной деятельности студентов должна удовлетворять следующим требованиям: наличие веб-интерфейса; возможность размещения студентами информации по учебным вопросам, прежде всего гиперссылок на открытые образовательные ресурсы; хранение данных о оценке ресурса, комментариях и рейтинге поль-

зователя в базе данных; возможность поиска записей; наличие личных кабинетов; поддержка общения пользователей; возможность добавления записей в закладки.

Диаграмма прецедентов – графическая диаграмма, которая отображает действующие лица информационной системы (прежде всего пользователей), варианты использования системы и связи между ними. На основе анализа требований к разрабатываемым компонентам информационной системы были выявлены следующие прецеденты – основные функции, доступные пользователям системы. Прецеденты (варианты использования) и пользователи, которым доступны эти функции, показаны на рис. 1 в соответствии со стандартом языка UML [4].

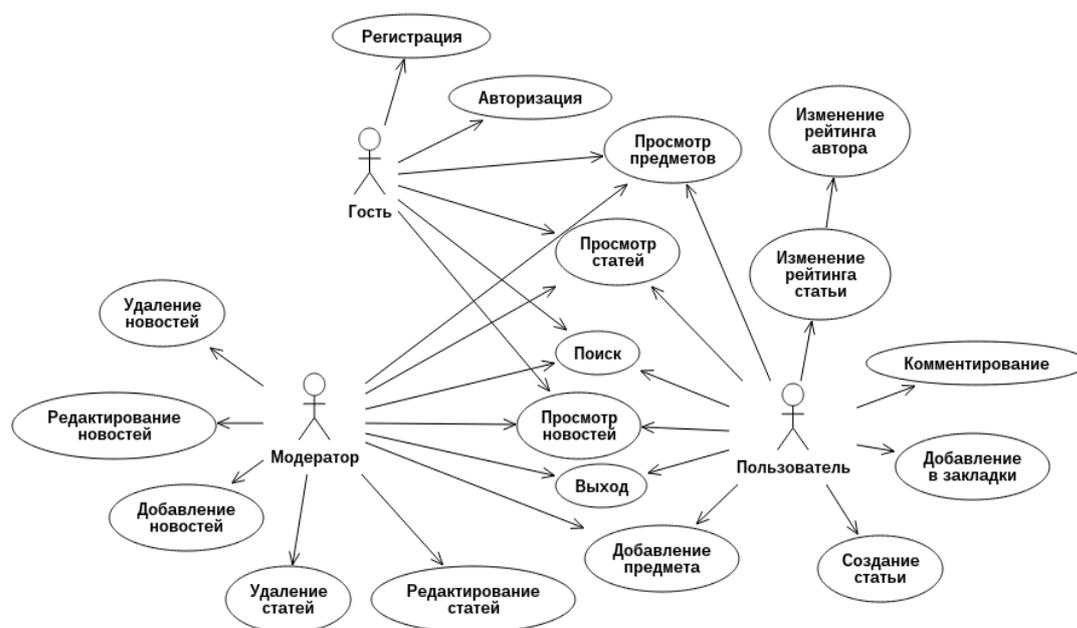


Рис. 1. Диаграмма прецедентов

На основе анализа предметной области и требований к ИС было выполнено проектирование базы данных. Предметная область описывает данные пользователей, учебных материалов, областях знаний, к которым относится материал, дисциплинах и темах, оценках и комментариях к образовательным ресурсам, рейтингах ресурсов и пользователях, закладках в личном кабинете. В результате работы был разработан прототип ИС с использованием СУБД MySQL, языка PHP и фреймворка Yii.

Библиографический список

1. Бершадский, А. М. Выполнение требований ФГОС 3+ – шаг в развитии электронного обучения / А. М. Бершадский, Т. В. Глотова, И. Г. Кревский // Информационное общество: образование, наука, культура и технологии будущего : тр. XVIII объединенной конф. «Интернет и современное общество» (IMS-2015) (23–25 июня 2015 г., Санкт-Петербург). – СПб. : Университет ИТМО, 2015. С. 21–32.
2. Бершадский, А. М. Современный университет: образовательные и информационные технологии в многоуровневой системе высшего образования / А. М. Бершадский, Т. В. Глотова, И. Г. Кревский // Информационное общество: образование, наука, культура и технологии будущего: тр. XVIII объединенной конф. «Интернет и современное общество» (IMS-2016) (21–24 июня 2016 г., Санкт-Петербург). – СПб. : Университет ИТМО, 2016. – С. 143–160.
3. Бершадский, А. М. Электронная информационно-образовательная среда организации как основа дальнейшего развития электронного обучения / А. М. Бершадский, И. Г. Кревский, В. А. Мещеряков // Научно-образовательная информационная среда XXI века : материалы IX Всерос. науч.-практ. конф. (23–25 сентября 2015 г.). – Петрозаводск, 2015. – С. 16–20.
4. Язык UML. Руководство пользователя / Г. Буч, Д. Рамбо, И. Якобсон. – М. : ДМК Пресс, 2008. – 496 с.

Л. В. Гурьянов, Е. А. Дзюба, Д. А. Миронов

Пензенский государственный университет, Пенза, Россия

РАЗРАБОТКА СТУДЕНЧЕСКИХ КУРСОВ НА ПЛАТФОРМЕ SOFTWARE-ACADEMY

Рассматривается методика разработки преподавателем студенческих учебных курсов на платформе Software-Academy.

С целью повышения качества подготовки выпускников на кафедре «Математическое обеспечение и применение ЭВМ» начата апробация учебного процесса на основе Git-технологии. В учебном процессе кроме преподавателей и студентов участвуют специалисты фирмы «Открытые решения» (Россия, Пенза). Сотрудничество осуществляется в рамках проекта «Software-Academy».

Предлагаемая схема процесса выполнения и проверки студенческих заданий (workflow) показана на рис. 1. Для организации совместной работы студентов, преподавателей и специалистов фирмы используется современная платформа GitLab, которая обеспечивает:

- сборку проекта из исходников студенческого решения;
- разворачивание/установка проекта на сервере;
- тестирование проекта;
- создание отчета и уведомление студента и преподавателя о результатах.

Кафедра, согласно учебному плану, осуществляет проведение аудиторных занятий, постановку задач, web-мониторинг прохождения задач студентами на сервере проекта, контроль и прием решений студентов.

Студенты в рамках самостоятельной работы отправляют свои решения на сервер проекта для его дальнейшей автоматизированной обработки.

Фирма обеспечивает автоматизированный сбор и проверку студенческих решений, анализ качества кода и консультации студентов специалистами фирмы.

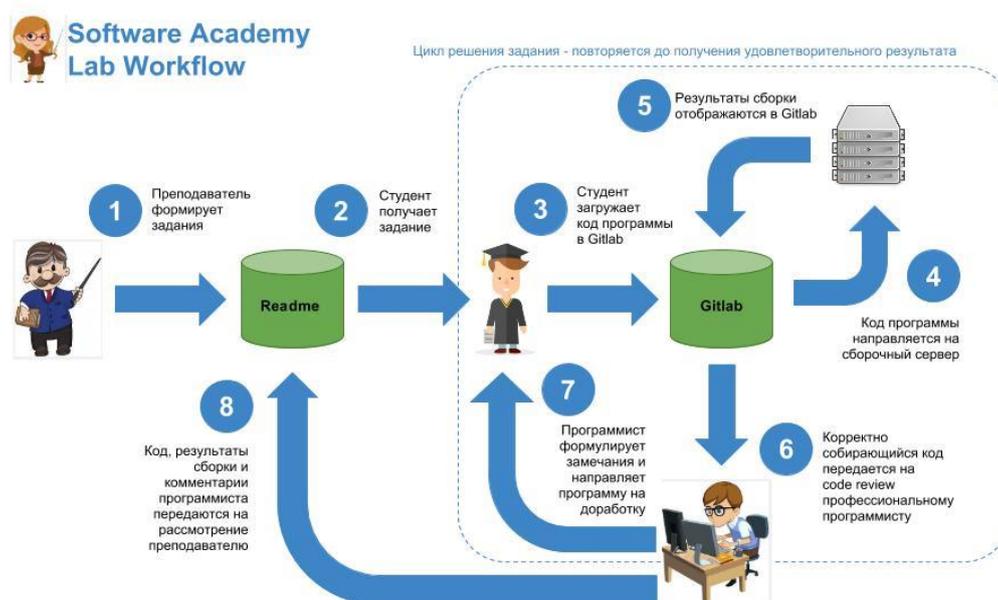


Рис. 1. Схема учебного процесса на платформе Software-Academy

Первым и наиболее трудоемким шагом реализации предлагаемого учебного процесса является разработка преподавателем компьютеризированного курса по изучаемой студентами дисциплине. Рассмотрим подробнее действия преподавателя в этом процессе.

GitLab предоставляет возможность создавать иерархии проектов-репозиториях. Это похоже на логическую схему файловой системы: подгруппа GitLab – это подкаталог, проект GitLab – это файл. Для доступа к каждой группе и любому проекту GitLab существует свой «путь».

На верхнем уровне иерархии определена одна или несколько групп типа «Группа поставщика курса». На рис. 2 – это группа «OpenSolutions»(фирма «Открытые решения») и группа «PSU» (Пензенский государственный университет).

GitLab-подгруппа «Направление обучения» (название одной или нескольких родственных учебных дисциплин) определяется на усмотрение поставщика курса – преподавателя, например, «Программирование на языке C++» (на рис. 2 – «Cpp»). Если изучение дисциплины происходит в студенческих группах разных направлений, следует добавить GitLab-подгруппы в подгруппу изучения дисциплины. На кафедре МО и ПЭВМ ПГУ готовят бакалавров по двум направлениям: «Программная инженерия» и «Информационные системы и технологии», поэтому в подгруппу«Cpp» будут добавлены GitLab-подгруппы «VP» и «VI» в соответствии с обозначением специальностей по этим направлениям.



Рис. 2. Иерархия групп поставщиков курсов

При создании GitLab-подгруппы в поле «Description» вносится необходимая информация об учебной дисциплине (количество зачетных единиц, общее количество часов, а также количество лабораторных работ и другая информация). После формирования новых курсов следует обязательно проверить актуальность информации в этом поле.

После создания иерархии GitLab-подгрупп необходимо создать шаблон учебной дисциплины – шаблон-проект GitLab. На основе шаблона будет создаваться учебный проект для студента (или для группы студентов в случае командной работы над одним проектом). В процессе создания шаблона преподаватель выполняет следующие действия:

1. Создание нового проекта:
 - на странице «New project» вводит название проекта («Project Path») и краткое описание проекта («Project Description»);
2. Добавление описания созданного проекта:
 - на открывшейся странице созданного проекта добавляет полное описание проекта («Add Readme»). В описание следует включить общие сведения о проекте, требования к

выполнению лабораторных работ, общее описание лабораторных работ, названия веток, которые студент должен создать для выполнения лабораторных работ, ссылки на инструкции по работе в GitLab и другую необходимую для студента информацию;

- сохраняет описания в репозитории проекта (GitLab «Repository»). Рекомендуется в репозитории размещать и PDF-файл с описанием лабораторных работ;

3. Добавление заданий на лабораторные работы:

- создает новую задачу («Issues»);
- добавляет описание заданий на конкретную лабораторную работу;
- сохраняет задачу («Submit issue»).

Таким образом, при наличии разработанных методических средств обучения по конкретной дисциплине переход на платформу «Software-Academy» выполняется достаточно просто, хотя и требует определенных трудозатрат.

После создания преподавателем иерархии GitLab-подгрупп и шаблона проекта по изучаемой дисциплине студенты клонируют этот шаблон в создаваемую ими ветку репозитория и выполняют указанные задания. Учебный процесс переходит на интерактивную электронную платформу, что делает его привлекательным для студентов, а также помогает им освоить современные технологии разработки программных средств.

А. Г. Мирзалиев

Ростовский государственный экономический университет, Ростов-на-Дону, Россия

WEB-ОРИЕНТИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ РАБОТЫ ПРЕПОДАВАТЕЛЕЙ ВУЗА

Представлена web-ориентированная система оценки эффективности деятельности преподавателей вуза, позволяющая в оперативном режиме вести учет и анализ достижений преподавателей на основе анкетирования в режиме онлайн. Автоматизированные процедуры формирования рейтинга ППС обеспечивают руководству вуза информационную поддержку для принятия решений по стимулированию профессиональной активности ППС.

Повышение качества системы высшего профессионального образования тесно связано с совершенствованием системы подготовки и аттестации преподавателей вузов. В целях повышения качества образовательных услуг необходимо внедрять систему оценки качества деятельности ППС. Рассматривались подходы к решению проблемы повышения качества образования посредством совершенствования учебных планов и программ, внедрения новых форм управления, использования технических средств, но такие попытки не привели к существенным сдвигам. Итогом проведенных исследований по вопросам повышения качества образования и выявлению основных влияющих факторов на эффективность образовательного процесса, стал вывод о том, что главным фактором является «человеческий», и осознании того факта, что оценка качества педагогических кадров и использование полученных в результате их анализа результатов является эффективным инструментом мотивации персонала [1].

При создании эффективной системы оценки качества преподавателей возникает необходимость разработки моделей и методов оценки качества деятельности педагогических кадров и их внедрение в систему управления образовательным процессом вуза с использованием современных информационных технологий. Создание web-ориентированной информационной системы, позволяющей в режиме онлайн вести оперативный учет и анализ деятельности преподавателей в разрезе выполненных работ позволяет создать базу данных, которую можно использовать как при решении вопросов стимулирования сотруд-

ников, так и для формирования рекомендаций по совершенствованию форм и методов управления ППС.

Ускоренные темпы развития рыночной экономики требуют повышения качества образовательного процесса в каждом вузе. Перед руководством вуза стоит задача, как обеспечить внедрение инноваций в учебный процесс, повысить уровень квалификации преподавателей, как мотивировать преподавателей вуза. Эффективным средством повышения качества работы ППС является ее рейтинговая оценка по основным видам деятельности. Вузы в настоящее время внедряют отдельные методики для оценки работы преподавателей и структурных подразделений в вузе. Однако полностью автоматизированных систем внедрено немного [2].

В разработанной нами информационной системе оценка деятельности ППС происходит по направлениям: учебная работа; научно-исследовательская деятельность; воспитательная работа; организационная и общественная деятельность. Всем видам работ, представленным в форме анкеты рейтинга руководством вуза поставлен удельный вес в баллах. В процессе заполнения анкеты каждым преподавателем в режиме онлайн по результатам выполненных работ, автоматически выполняется подсчет баллов. В результате расчетов формируется итоговая суммарная оценка в баллах, которая определяет все достижения преподавателя по всем видам его деятельности. Одновременно формируется общий рейтинг ППС в вузе, рейтинг факультетов и кафедр. Результаты обработки в виде отчетов предоставляются руководству вуза.

На рис. 1 приведена онлайн форма анкеты преподавателя.

АНКЕТА
рейтинговой оценки преподавателя
Рыбкина Алена Сергеевна кафедры КИЛ
факультета ТД
за 2017 год

Дата создания анкеты 2017-11-17 11:12:13

550.51 30.00 30.00 10.00 22.8% 620.51
НИР УМР ВР ВОВ ПРОЦЕНТ ОБЩИЙ БАЛЛ

| №№ п/п | Наименование показателя | Единица измерения факт | Периодичность измерения | Весомость показателя (в баллах) | Документ, подтверждающий выполнение показателя | Итоговое значение показателя в баллах | Файлы |
|---|--|------------------------|-------------------------|---------------------------------|--|---------------------------------------|-------|
| I Научно-исследовательская и инновационная деятельность | | | | 2017 | | | |
| 1 | Монографии, изданные издательствами (не менее 500 экземпляров) (соавторы в долях авторских п.л.), в том числе: | | | | | | |
| | - зарубежными | 0 | 1 календ. год | 40 | Предоставление в библиотеку РГЭУ (РИНХ) не менее о ... | 0 | |
| | - центральными | 1.28 | 1 календ. год | 37.00 | Предоставление в библиотеку РГЭУ (РИНХ) не менее о ... | 47.36 | |
| | - региональными, вузовскими | 0.33 | 1 календ. год | 35.00 | Предоставление в библиотеку РГЭУ (РИНХ) не менее о ... | 11.55 | |
| 2 | Публикации в изданиях, индексируемые в информационно-аналитических системах (соавторы в долях авторских п.л.): | | | | | | |
| | - Web of Science: квартал 1-2 | 0 | 1 календ. год | 150 | Скан-копия выходных данных публикации и библиометр ... | 0 | |
| | - Web of Science: квартал 3-4, в т.ч. нулевой | 0 | 1 календ. год | 100 | Скан-копия выходных данных публикации и библиометр ... | 0 | |

Рис. 1. Анкета рейтинга преподавателя кафедры

Основными функциями разработанной web-ориентированной информационной системы являются:

- ведение базы данных по достижениям сотрудников кафедры в виде анкет по каждому отчетному периоду (месяц, семестр, учебный год, несколько лет) для предоставления по требованию руководству; формирование рейтинга преподавателей по кафедре, факультету и вузу в целом;
- формирование рейтинга кафедр, факультетов вуза;
- автоматизация процессов формирования отчетов в текстовом и графическом формате по результатам деятельности сотрудников, подразделений анкете;
- ограничение и защита информации в зависимости от прав доступа каждого пользователя ИС.

При разработки были учтены следующие требования к созданию системы: реализация бизнес-логики системы в соответствии с выявленными требованиями и сценариями ра-

боты всех групп пользователей; требования юзабилити – удобство пользователя при работе с системой; быстрдействие системы; защита персональных данных; способность к модификации и адаптации в соответствии с изменяемыми требованиями [3]. На основе анализа требований, был разработан web-сервис для оценки результатов деятельности ППС, который подключен к образовательному portalу вуза.

Особенностью разработанной системы является наличие личного кабинета преподавателя и возможность онлайн-заполнения и просмотра результатов деятельности с подтверждающими документами. Входными данными для анкет является информация о выполненных работах по индивидуальному плану преподавателя за текущий период. В результате формируется единая база данных о деятельности преподавателей, на основе которой появляется возможность в дальнейшем составлять автоматически отчеты о деятельности каждого структурного подразделения за любой отчетный период по всем направлениям работы. Следовательно, нет необходимости отдельного заполнения фактических отчетов, а можно использовать информацию непосредственно из базы web-ориентированной информационной системы. На рис. 2 приведена одна из форм отчета по результатам деятельности преподавателей кафедры.



Рис. 2. Рейтинг преподавателей по кафедре

Созданная информационная система рейтинговой оценки позволяет вести оперативный учет выполненных работ каждым преподавателем по всем запланированным направлениям его деятельности. Сформированная база данных может быть использована для последующего анализа причин возникновения недостатков в работе и возможностей по их устранению путем корректировки планов работ, совершенствования учебно-методического обеспечения и научного потенциала вуза.

Библиографический список

1. Карпенко, М. П. Эволюция управления качеством и задачи его реализации в современном российском высшем образовании / М. П. Карпенко, В. Н. Фокина, В. А. Березовский, А. В. Слива // Инновации в образовании. – 2010. – № 2. – С. 21–38.
2. Архипова, Е. Н. Автоматизация рейтинговой оценки деятельности преподавателя / Е. Н. Архипова, О. В. Кононова, В. В. Крюков, К. И. Шахгельдян // Университетское управление: практика и анализ. – 2010. – № 5. – С. 51–62.
3. Шполянская, И. Ю. Автоматизированные процедуры поддержки принятия решений для определения рациональной структуры информационной системы управления предприятием малого бизнеса / И. Ю. Шполянская // Приборы и системы. Управление, контроль, диагностика. – 2005. – № 7. – С. 63–66.

Пензенский государственный университет, Пенза, Россия

МЕТОДИКА АНАЛИЗА АЛГОРИТМОВ ОБРАБОТКИ ДАННЫХ

Посвящена описанию программного стенда для анализа алгоритмов обработки данных. Стенд предназначен для сравнения альтернативных алгоритмов обработки данных на ранних стадиях проектирования программного обеспечения с целью выбора наиболее эффективного алгоритма.

Традиционный способ сравнения эффективности алгоритмов состоит в сопоставлении их порядков сложности. Этот метод применим как к временной, так и пространственной сложности. Порядок сложности алгоритма выражает его эффективность обычно через количество обрабатываемых данных.

Таким образом, в теории алгоритмов широко используются и повсеместно применяются методы сравнительного анализа алгоритмов, основанные на понятии теоретической сложности алгоритмов, т.е. O -оценок.

Однако такой подход к оценке эффективности алгоритмов имеет ряд недостатков. К основным недостаткам подхода можно отнести следующие:

- 1) для сложных алгоритмов получение O -оценок, как правило, либо очень трудоемко, либо практически невозможно;
- 2) часто трудно определить сложность «в среднем»;
- 3) O -оценки слишком грубые для отображения более тонких отличий алгоритмов;
- 4) O -анализ дает слишком мало информации (или вовсе ее не дает) для анализа поведения алгоритмов при обработке небольших объемов данных.

В частности, одним из основных недостатком O -функций является их чрезмерная грубость, неточность [1, 2]. Если алгоритм A выполняет некоторую задачу за $0.001 \times n$ (n – размерность обрабатываемых данных) временных единиц, в то время как для ее же решения с помощью алгоритма B требуется $1000 \times n$ временных единиц, то алгоритм A в миллион раз быстрее, чем алгоритм B . Тем не менее алгоритм A и алгоритм B имеют одну и ту же теоретическую временную сложность $O(n)$.

Наиболее известные алгоритмы внутренней сортировки: сортировка с помощью прямого обмена (пузырек), сортировка с помощью прямого выбора и сортировка с помощью прямого включения имеют теоретическую сложность $O(n^2)$, в то время как практическая сложность этих алгоритмов может отличаться в разы.

Таким образом, основная проблема при использовании традиционного подхода к анализу сложности алгоритмов, заключается в том, что он зачастую не дает ответа на главный вопрос, какой из возможных алгоритмов решения задачи является наиболее эффективным. Следовательно, часто необходим более детальный анализ алгоритмов, претендующих на решение одной и той же задачи.

В связи с этим разработка новых методик сравнительного анализа алгоритмов обработки данных является в настоящее время очень актуальной.

В работе для решения поставленной выше проблемы предлагается программный стенд для анализа алгоритмов обработки данных, который позволяет для заданных в программе исходных данных проанализировать практическую эффективность наиболее известных алгоритмов решения задачи, имеющих одинаковую теоретическую сложность [3].

Входными данными стенда являются: входные параметры обрабатываемых данных (тип, размер, степень упорядоченности, требования к устойчивости и т.п.) и перечень сравниваемых алгоритмов их обработки.

Выходные данные – табличный и визуальный анализ использования предложенных алгоритмов при заданных исходных данных, рекомендации по выбору алгоритма обработки данных при заданных исходных данных.

В настоящее время разработан стенд для анализа алгоритмов внешней сортировки [4] и стенд для анализа алгоритмов внутренней сортировки. В статье будут рассмотрены основные возможности использования стенда для анализа алгоритмов внутренней сортировки (рис. 1).

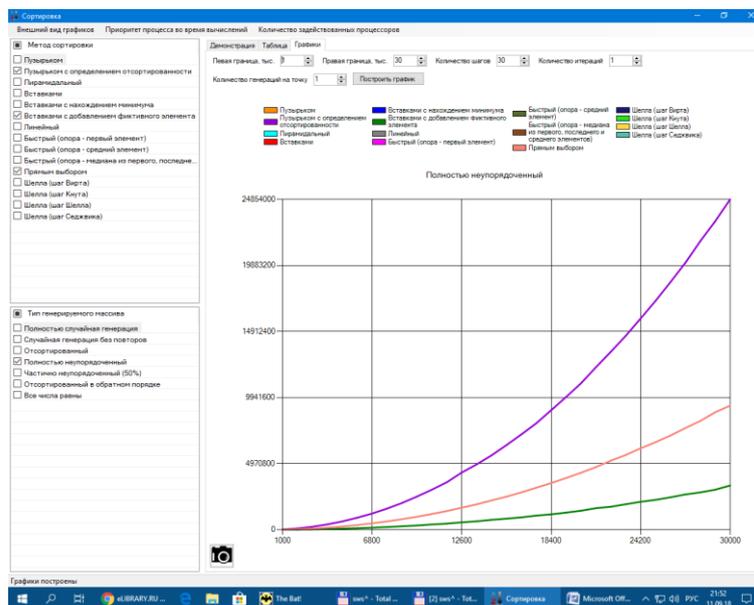


Рис. 1. Программа для анализа алгоритмов внутренней сортировки

Программа позволяет выполнять сравнение любого сочетания из пятнадцати наиболее известных алгоритмов внутренней сортировки. Кроме того, выбранные алгоритмы могут быть проанализированы для семи различных типов начальной упорядоченности исходного массива (неупорядоченного, упорядоченного, упорядоченного в обратном порядке, частично упорядоченного и т.д.).

Для генерации визуального представления сравниваемых алгоритмов программа позволяет задавать следующие входные параметры: размерность массива и количество точек, для которых рассчитываются значения времени сортировки.

Кроме того, для сглаживания получаемых графиков используются еще два параметра: количество итераций и количество генераций на точку.

Разрабатываемый проект имеет в первую очередь практическое применение. Как было показано выше, задача выбора наиболее эффективного алгоритма обработки данных из множества альтернативных является одной из наиболее важных на этапе анализа поставленной задачи.

Предлагаемый программный стенд позволит уже на начальных этапах разработки программных систем выбирать именно те алгоритмы обработки данных, которые наиболее эффективны для заданных исходных данных и условий решения задачи.

Библиографический список

1. Самуйлов, С. В. Методика сравнительного анализа алгоритмов на примере алгоритмов последовательного поиска / С. В. Самуйлов // Концепт. – 2014. – № 9. – С. 46–50.
2. Самуйлов, С. В. Сравнение и анализ алгоритмов последовательного поиска / С. В. Самуйлов // Управление реформированием социально-экономического развития предприятий, отраслей, регионов : материалы IV Междунар. науч.-практ. конф. – Пенза, 2013. – С. 262–265.
3. Самуйлов С. В. Современные информационные технологии в обучении студентов-заочников / С. В. Самуйлов // Концепт. – 2014. – № 8. – С. 111–115.
4. Самуйлов, С. В. Анализ алгоритмов внешней сортировки / С. В. Самуйлов, А. О. Денисов, В. Ю. Дунаев // Управление реформированием социально-экономического развития предприятий, отраслей, регионов : сб. науч. тр. IX Междунар. науч.-практ. конф. студентов, аспирантов, преподавателей и практиков. – Пенза, 2018. – С. 268–269.

Пензенский государственный университет, Пенза, Россия

ОБЕСПЕЧЕНИЕ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ ТЕХНОЛОГИЯМИ «1С» ПРИ ПОДГОТОВКЕ БАКАЛАВРОВ ПО ПРОФИЛЮ «ПРИКЛАДНАЯ ИНФОРМАТИКА В ЭКОНОМИКЕ»

Рассматривается поддержка формирования профессиональных компетенций технологиями «1С» при подготовке бакалавров по прикладной информатике в экономике.

Прикладная информатика, как направление профессиональной подготовки, объединяет подготовку в области информационных технологий и в той предметной области, где эти технологии используются, в данном случае – в экономике и финансах. Бакалавр прикладной информатики в экономике должен быть профессионалом, обладающим фундаментальными и прикладными знаниями, соответствующими компетенциями, как в области информатики, так и в области экономики. Целью подготовки таких профессионалов является готовность к решению задач создания и эксплуатации экономических информационных систем прикладного назначения, а также автоматизации управленческой деятельности, овладению профессиональными компетенциями, способствующими социальной мобильности и устойчивости на рынке труда.

Фундамент подготовки бакалавров по прикладной информатике в экономике – «Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.03 Прикладная информатика (уровень бакалавриата)» [1].

Стандарт обеспечивает единство образовательного пространства РФ, качество образования; служит основой для объективной оценки деятельности вузов; позволяет устанавливать эквивалентность документов об образовании других стран.

Главной особенностью ФГОС является компетентностный подход с ориентацией на результат обучения и на цели воспитания и обучения. Основными принципами компетентностно-ориентированных образовательных программ являются:

- Фокусировка на задачах из профессиональной практики.
- Интеграция усвоения и применения знаний и навыков.
- Персональная ответственность студента.
- Обучение в сотрудничестве.
- Новые формы оценки.
- Использование информационно-коммуникационных технологий.
- Развитие компетенций обучаемого для его профессиональной карьеры

Одним из трех видов компетенций, указанных в стандарте являются профессиональные компетенции. Профессиональные компетенции – это способности или обязанности работника выполнять поставленные задачи в соответствии со стандартами их выполнения. Технологии и программные продукты фирмы «1С» как нельзя лучше подходят для формирования профессиональных компетенций бакалавров профиля «Прикладная информатика в экономике».

ПК-2 – способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение.

«1С:Предприятие 8» состоит из совокупности программ. Этот комплекс помогает решать многочисленные задачи автоматизации и учета на производстве. Каждое предприятие имеет свои потребности и характеристики. Например, размер производства, область применения продукции, а также структура управления предприятием и т.д. Именно поэтому в системе программ «1С:Предприятие» есть возможность создавать индивидуальную информационную систему производства, учитывая всю его специфику. Внедрение и сопровождение осуществляют, как и IT-специалисты отдельных фирм, так и сами партнеры

«1С». Все эти особенности представляет прогрессивная архитектура «1С». То есть, технологическая платформа и, основанные из ней, прикладные решения.

ПК-4 – способность документировать процессы создания информационных систем на стадиях жизненного цикла.

Жизненный цикл информационных систем имеет несколько стадий: постановка, проектирование, разработка и развертывание, ввод в действие, модернизация или ликвидация. На платформе «1С:Предприятие» система проектирование прикладных решения (СППР) используется для проектирования прикладных решений, а также ведения технической документации проекта. СППР также может использоваться в виде инструмента для проектирования новых информационных систем, которые разрабатываются в среде «1С:Предприятия 8» и для описания и документирования систем, созданным ранее без использования СППР.

ПК-10 – способность принимать участие во внедрении, адаптации и настройке информационных систем.

Внедрения прикладных решений осуществляются благодаря партнерам-внедренцам. Так как этот порядок является наиболее удобным, то это позволяет наиболее точно организовывать пожелания предприятий. Фирма «1С» ведет справочник внедренных решений, который состоит из информационных сообщений партнеров о внедренных решениях на основе системы программ "1С:Предприятие. Внедрения могут осуществлять, как и IT-специалисты других компаний, там и фирмы «1С». Опыт внедрения прикладных решений на платформе 1С:Предприятие 8 показывает, что система позволяет решать задачи различной степени сложности.

ПК-11 – способность эксплуатировать и сопровождать информационные системы и сервисы.

Состав программ системы «1С:Предприятие» ориентирован на актуальные потребности предприятий. Фирма «1С» выпускает тиражные программные решения, предназначенные для автоматизации типовых задач учета и управления в коммерческих предприятиях реального сектора и бюджетных организациях. В каждом программном продукте сочетается использование стандартных решений и максимальный учет специфики задачи конкретной отрасли или рода деятельности предприятия. Также фирма «1С» осуществляет выпуск стандартного набора решений, предназначенного для наиболее широкого спектра предприятий. Фирмы, которые занимаются разработкой и распространением отраслевых решений на платформе «1С:Предприятие», создают специализированное тиражное решение. В процессе этого они используют методологические решения, разработанные «1С». Состав программ «1С:Предприятия» и набор их функции динамично развиваются вместе с изменением типовых потребностей отечественных предприятий и организаций.

ПК-14 – способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач.

Система «1С: Предприятие» поддерживает два варианта работы с информационной базой: файловый и клиент-серверный. Файловый вариант работы рассчитан на персональную работу одного пользователя или работу небольшого количества пользователей в локальной сети. В этом варианте все данные информационной базы (конфигурация, база данных, административная информация) располагаются в одном файле. Клиент-серверный вариант предназначен для использования на средних и крупных предприятиях. Он реализован на основе трехуровневой архитектуры «клиент-сервер».

ПК-22 – способность анализировать рынок программно-технических средств, информационных продуктов и услуг для создания и модификации информационных систем.

Система программ «1С:Предприятие» предназначена для автоматизации управления и учета на предприятиях различных отраслей, видов деятельности и типов финансирования, и включает в себя решения для комплексной автоматизации производственных, торговых и сервисных предприятий, продукты для управления финансами холдингов и отдельных предприятий, ведения бухгалтерского учета («1С:Бухгалтерия» самая известная учетная программа в ряде стран), расчета зарплаты и управления кадрами, для учета

в бюджетных учреждениях, разнообразные отраслевые и специализированные решения, разработанные самой фирмой «1С», ее партнерами и независимыми организациями.

Подводя итоги, можно утверждать, что комплекс программ «1С» позволяет формировать наиболее важные компетенции бакалавров прикладной информатики в экономике.

Таким образом, технологии 1С позволяют формировать наиболее важные профессиональные компетенции бакалавров профиля подготовки «Прикладная информатика в экономике».

Библиографический список

1. Федеральный государственный образовательный стандарт высшего образования. Уровень высшего образования – бакалавриат. Направление подготовки – 09.03.03 : [утв. приказом Министерства образования и науки РФ № 207 от 12 марта 2015 г.].

А. В. Ширканов

Пензенский государственный университет, Пенза, Россия

ПЕРВЫЕ РЕЗУЛЬТАТЫ ИСПОЛЬЗОВАНИЯ КОМПОНЕНТА КОНТРОЛЯ НАПОЛНЕНИЯ УЧЕБНЫХ КУРСОВ ЭИОС

Разработан и успешно применен компонент контроля наполнения учебных курсов.

Новые федеральные государственные образовательные стандарты (ФГОС) требуют применения информационно-коммуникационных технологий в процессе реализации образовательных программ [1], что порождает, в том числе огромное количество электронных образовательных ресурсов [2]. В применяемой электронной информационной образовательной среде (ЭИОС) ПГУ разработан и успешно применен компонент контроля наполнения учебных курсов.

Компонент позволяет в режиме on-line отслеживать процесс наполнения и настройки образовательных программ по всем направлениям (специальностям) бакалавриата, специалитета, магистратуры, аспирантуры, ординатуры и СПО, а так же ведения образовательного процесса в организации. В результате использования ЭИОС, кроме выявления незаполненных обязательных составляющих в каждом учебном курсе (согласно положениям постановления Правительства Российской Федерации от 10.07.2013 № 582) были выделены следующие потребности:

- в выявлении учебных курсов, по которым не ведется форум, не настроена балльно-рейтинговая система;
- в получении информации об учебных планах, в которых не настроены дисциплинах по выбору, а так же о «привязанных» к таким дисциплинам академических групп;
- в получении данных о группах и их преподавателях, по которым не выставляется посещаемость или аттестация;
- в подсчете количества незаполненных тем курсовых или выпускных квалификационных работ.

Исходя из озвученных потребностей, был составлен плана-график на разработку и внедрение дополнительных функций, обеспечивающих данные возможности. На текущий момент в компонент добавлен функционал по выявлению ведения форума, как элемента взаимодействия между участниками образовательного процесса.

В табл. 1 приведено текущее состояние наполненности системы (по состоянию на момент написания статьи).

Таблица

Наполненность ЭИОС

| Уровень | Число курсов в уровне | Тип курсов | Число курсов в типе |
|------------------------|-----------------------|--|---------------------|
| Всего загружено | 29694 | ПУСТЫЕ (содержит курсы без ресурсов или только с одним любым ресурсом) | 569 |
| | | БЕЗ АННОТАЦИИ ИЛИ РАБОЧЕЙ ПРОГРАММЫ (содержит курсы, в которых в первой или второй темах первый ресурс не файл) | 70 |
| | | ПОЛНОСТЬЮ/ЧАСТИЧНО ЗАПОЛНЕННЫЕ (содержит курсы, не попавшие в первые два раздела) | 29055 |
| | | ФОРУМ НЕ ВЕДЕТСЯ (содержит курсы, форумы которых содержат менее двух сообщений) | 28942 |
| | | ФОРУМ ИСПОЛЬЗУЕТСЯ (содержит курсы, форумы которых содержат два и более сообщений) | 752 |
| Из них | | | |
| <u>Бакалавриат</u> | 18875 | пустые | 267 |
| | | без аннотации или рабочей программы | 26 |
| | | полностью/частично заполненные | 18582 |
| | | форум не ведется | 18375 |
| | | форум используется | 500 |
| <u>Магистратура</u> | 3140 | пустые | 155 |
| | | без аннотации или рабочей программы | 5 |
| | | полностью/частично заполненные | 2980 |
| | | форум не ведется | 3052 |
| | | форум используется | 88 |
| <u>Ординатура</u> | 495 | пустые | 0 |
| | | без аннотации или рабочей программы | 1 |
| | | полностью/частично заполненные | 494 |
| | | форум не ведется | 494 |
| | | форум используется | 1 |
| <u>СПО</u> | 1782 | пустые | 8 |
| | | без аннотации или рабочей программы | 35 |
| | | полностью/частично заполненные | 1739 |
| | | форум не ведется | 1771 |
| | | форум используется | 11 |
| <u>Аспирантура</u> | 1766 | пустые | 0 |
| | | без аннотации или рабочей программы | 1 |
| | | полностью/частично заполненные | 1765 |
| | | форум не ведется | 1766 |
| | | форум используется | 0 |
| <u>Специалитет</u> | 3636 | пустые | 139 |
| | | без аннотации или рабочей программы | 2 |
| | | полностью/частично заполненные | 3495 |
| | | форум не ведется | 3484 |
| | | форум используется | 152 |

Детальный просмотр информации по образовательной программе приведен на рис. 1.

| Курс | Название дисциплины | Подразделение | Ответственный | Назначенные преподаватели | Ресурсы по темам |
|--------|---|---|---|--|---|
| 1 курс | всего дисциплин выбранного типа в курсе: 13 | | | | |
| | Английский язык | Кафедра "Английский язык" [АЯ] |  Мещерякова Елена Александровна  dance_lina@mail.ru | 1. Данкова Наталья Станиславовна 2. Бехтер Анна Юрьевна 3. Видниеева Наталья Юрьевна 4. Видниеева Наталья Юрьевна | <input type="button" value="▼"/> Всего ресурсов в курсе: 11 |
| | Геоинформационные технологии и системы | Кафедра "Системы автоматизированного проектирования" [САПР] |  Бурукина Ирина Петровна  burukinairina@gmail.com | 1. Бождай Александр Сергеевич | <input type="button" value="▼"/> Всего ресурсов в курсе: 15 |
| | Информационные системы управления | Кафедра "Системы автоматизированного проектирования" [САПР] |  Бурукина Ирина Петровна  burukinairina@gmail.com | 1. Бождай Александр Сергеевич | <input type="button" value="▼"/> Всего ресурсов в курсе: 15 |
| | Методология и технология проектирования информационных систем | Кафедра "Системы автоматизированного проектирования" [САПР] |  Бурукина Ирина Петровна  burukinairina@gmail.com | 1. Глотова Татьяна Владимировна | <input type="button" value="▼"/> Всего ресурсов в курсе: 39 |
| | Метрология качества программного обеспечения | Кафедра "Системы автоматизированного проектирования" [САПР] |  Бурукина Ирина Петровна  burukinairina@gmail.com | 1. Селиверстова Ирина Александровна | <input type="button" value="▼"/> Всего ресурсов в курсе: 6 |
| | Научно-исследовательская работа | Кафедра "Системы автоматизированного проектирования" [САПР] |  Бурукина Ирина Петровна  burukinairina@gmail.com | 1. Бершадский Александр Моисеевич | <input type="button" value="▼"/> Всего ресурсов в курсе: 7 |
| | Непрерывные и дискретные математические модели | Кафедра "Компьютерные технологии" [КТ] |  Горонова Марина Юрьевна  gomj@mail.ru | 1. Яремко Олег Эмануилович | <input type="button" value="▼"/> Всего ресурсов в курсе: 11 |
| | Планирование и организация научных исследований | Кафедра "Информационно-вычислительные системы" [ИВС] |  Косников Юрий Николаевич  kosnikov@gmail.com | 1. Кузнецова Марина Владимировна | <input type="button" value="▼"/> Всего ресурсов в курсе: 22 |

Рис. . Детальный просмотр информации

Данный вариант просмотра повышает наглядность наполненности учебных курсов образовательной программы. Вдобавок в компоненте используется цветовая схема: зеленый – все хорошо, заполнены все обязательные компоненты; желтый – требует внимания; красный – ничего не заполнено, все обязательные компоненты пусты.

Особый интерес компонент вызывает у руководящего звена, хотя доступ к данному компоненту имеет каждый зарегистрированный пользователь ЭИОС. Компонент позволяет выявить недостающие элементы учебного курса. Для этого достаточно воспользоваться кнопкой рядом со строкой «Всего ресурсов в курсе». Отобразившаяся таблица, приведенная на рис. 2, покажет существующие и недостающие элементы.

| № | Название темы | Ресурсы [тип] |
|---|---|--|
| 0 | | <ul style="list-style-type: none"> Новостной форум (форум) |
| 1 | Аннотация программы дисциплины | <ul style="list-style-type: none"> Аннотация (файл) |
| 2 | Рабочая программа дисциплины | <ul style="list-style-type: none"> Рабочая программа дисциплины (файл) Лист изменений и переутверждений М1. 1.2 (файл) Рп 01.04.02 new (файл) |
| 3 | Документы, регламентирующие изучение дисциплины | <ul style="list-style-type: none"> ФЗ о науке и государственной научно-технической политике (файл) Гражданский кодекс. Часть 2 (файл) Гражданский кодекс. РФ. Часть 4 (файл) ГОСТ Р 55365. Научные произведения (файл) ГОСТ 7.32-2001 Отчет о НИР Структура и правила оформления (файл) ФГОС 01.04.02 (файл) |
| 4 | Учебно-методическое обеспечение дисциплины | <ul style="list-style-type: none"> Учебно-методическое пособие (файл) Методические рекомендации (файл) Мини-проект (файл) Рекомендации по подготовке публикаций (файл) |
| 5 | Дидактические материалы по дисциплине | <ul style="list-style-type: none"> Методика проведения патентного поиска (файл) Список международных научных наград (файл) |
| 6 | Оценочные средства по дисциплине | <ul style="list-style-type: none"> Задания, вопросы к зачету (файл) Тест (файл) |
| 7 | Образцы оформления отчетов | <ul style="list-style-type: none"> Лабораторная работа (файл) Презентация (файл) |
| 8 | Список литературы | <ul style="list-style-type: none"> Список литературы (файл) |

Рис.

После нахождения недостающих элементов в учебном курсе можно быстро принять управляющее воздействие на ответственного за наполнение сотрудника. Он указан в соседнем столбце данной таблицы. Там же приведен его контактный телефон и адрес электронной почты. Аналогичным образом можно проверить исполнение поручений по наполнению учебных курсов.

Библиографический список

1. Об образовании в Российской Федерации : федер. закон № 273-ФЗ от 29.12.2012.
2. Об утверждении Правил размещения на официальном сайте образовательной организации в информационно-телекоммуникационной сети Интернет и обновления информации об образовательной организации : постановление Правительства РФ № 582 от 10.07.2013.

**И. Г. Кревский, А. В. Антонов, М. В. Деев, Э. А. Зубринова,
И. А. Рензяева, А. В. Ширканов**

Пензенский государственный университет, Пенза, Россия

УПРАВЛЕНИЕ ВНЕДРЕНИЕМ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВУЗА

Рассмотрен состав организационных мероприятий при внедрении информационной системы вуза. На примере ЭИОС ПГУ рассмотрен набор инструментов, позволяющих осуществлять мониторинг и контроль внедрения.

Внедрение информационной системы может повысить эффективность деятельности организации любой сферы за счет следующих факторов:

- повышение эффективности основных бизнес-процессов;
- повышение скорости подготовки и оформления документов, экономия за счет минимизации бумажного документооборота;
- повышение качества управления за счет ускорения поиска информации, автоматизации ее сбора и аналитической обработки;
- сокращения расходов на подготовку документов и материалов по требованиям и запросам различных ведомств, организаций и регуляторов.

Все это в полной мере относится к внедрению в вузе электронной информационной образовательной среды (ЭИОС). Однако эффект может быть достигнут только в случае согласованной работы всех подразделений по использованию ЭИОС, своевременного занесения в систему необходимой информации.

При внедрении ЭИОС можно выделить организационные мероприятия и мониторинг использования системы. К организационным мероприятиям относятся следующие.

1. Принятие решения о разработке или адаптации системы и ее внедрении, разработка технического проекта.
2. Создание коллектива разработчиков или привлечение внешней компании-разработчика (при разработке ЭИОС ПГУ был создан проектный офис для разработки и внедрения системы [1]).
3. Формирование плана-графика разработки и внедрения системы.
4. Создание комиссии по приемке работ.
5. Регулярные заседания комиссии с отчетом разработчиков о полученных результатах и стадии внедрения (в ПГУ проводились ежемесячно на стадии разработки ЭИОС, на стадии сопровождения и доработки – раз в полгода).
6. Назначение ответственных за внедрение системы в подразделениях: помимо организационных моментов и помощи коллегам, на них возлагается взаимодействие с коман-

дой разработки (внедрения) и выполнение администраторских функций в пределах подразделения (в случае кафедр ПГУ – это закрепление дисциплин за преподавателями, настройка учебных планов и т.д.).

7. Обучение работе с системой. В процессе запуска в эксплуатацию ЭИОС ПГУ за полтора года повышение квалификации прошли почти 1500 человек (все преподаватели и работники деканатов, сотрудники ряда управлений). В ноябре 2017 года все лекции программы были записаны и размещены в специальном канале на YouTube.

8. Разработка локальных нормативных документов, регулирующих использование ЭИОС: Положение об ЭИОС, Положение об электронном портфолио обучающегося, Положение об электронном обучении и дистанционных образовательных технологиях.

9. Разработка методического обеспечения использования ЭИОС – разработано 12 инструкций, освещающих различные аспекты работы с ЭИОС для разных категорий пользователей.

10. Проведение организационно-методических семинаров. В последние недели перед вводом ЭИОС в эксплуатацию проводились еженедельно, в остальное время – в связи с наступлением определенного периода в учебной деятельности (например, начало очередного семестра, время подготовки дипломов и приложений к ним) либо выходом существенных обновлений ЭИОС (например, реализация балльно-рейтинговой системы (БРС), портфолио студента и т.д.). Проводятся либо для всех желающих, либо с ориентацией на конкретную категорию слушателей (ответственные за ЭИОС на кафедрах, работники деканатов, секретари ГАК и т.д.). Организованы интернет-трансляции семинаров, чтобы заинтересованные лица, которые не смогли прибыть на мероприятие, могли посмотреть его удаленно. Записи семинаров выкладываются в специальном канале на YouTube.

11. Система поддержки эксплуатации ЭИОС базируется на справочно-обучающем курсе в Moodle, где содержатся все нормативные и методические материалы, презентации, ссылки на видеоматериалы, тематические форумы поддержки «Форум для консультаций по загрузке ресурсов», «Вопросы регистрации пользователей», «Форум для консультаций по работе со студентами». Также организованы поддержка по e-mail и телефонные консультации.

12. Обучение студентов работе в ЭИОС – в основном возложено на преподавателей, проводящих вводное занятие в компьютерном классе в рамках изучения одной из учебных дисциплин. Частично были организованы поточные презентации системы для первокурсников с показом основных возможностей системы и приемов работы в ней. Разработана инструкция для студентов, размещены обучающие видеоролики.

13. Регулярное рассмотрение вопросов использования ЭИОС на ректорате, заслушивание вопроса на Ученом совете (как отдельного или как составной части развития информатизации вуза).

Очевидно, что работа по внедрению требует постоянного контроля и мониторинга, начиная с самоконтроля пользователей (проверка преподавателем, всю ли нужную информацию он разместил и т.д.), заканчивая контролем со стороны руководства вуза и подразделений. В связи с огромным объемом загружаемых ресурсов и данных в ЭИОС ручной мониторинг просто невозможен. Поэтому были разработаны методы автоматизированного контроля наполнения ЭИОС ресурсами и выполнения подразделениями обязанностей по отражению в ЭИОС своей деятельности. Разработанные инструменты мониторинга включают:

1) Компонент поиска незаполненных курсов <http://moodle.pnzgu.ru/checkup>. Его использование позволило оперативно отслеживать процесс первоначального наполнения ЭИОС, а также осуществлять мониторинг в процессе эксплуатации.

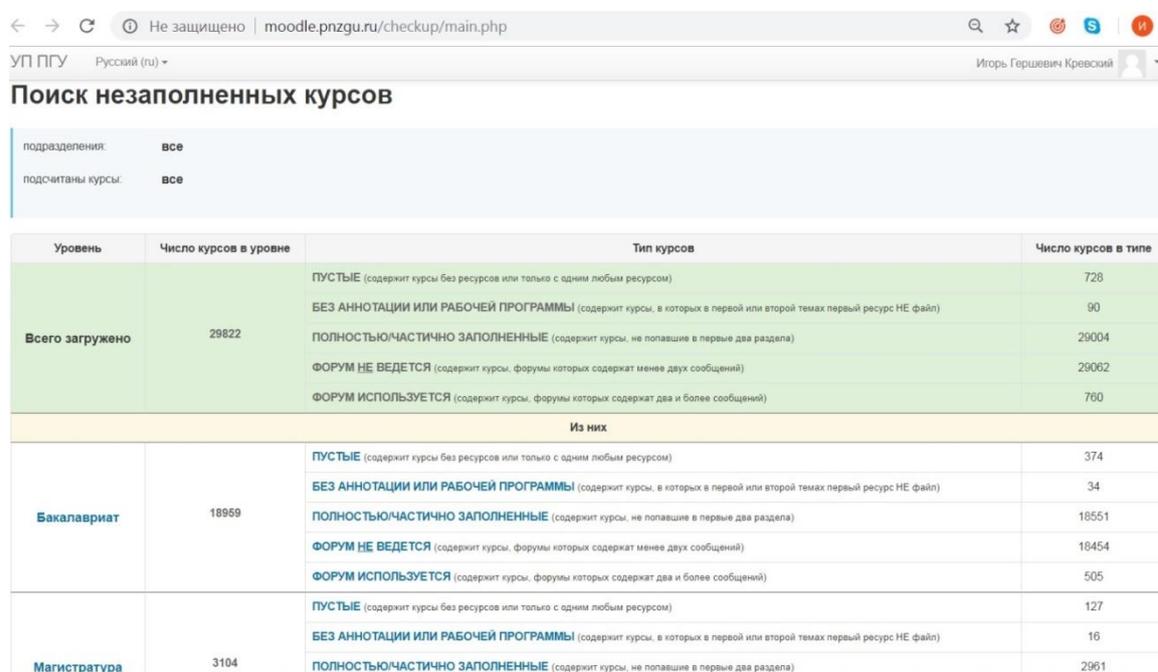
Доступ к компоненту имеют все зарегистрированные пользователи ЭИОС. Предусмотрены 3 режима просмотра:

– Полный список курсов – отображает наполнение курсов без выборки по отдельным подразделениям.

– Список курсов преподаваемых подразделением – отображает наполнение курсов, преподаваемых выбранными подразделениями (для факультетов и институтов – кафедрами, входящими в их состав).

– Список курсов преподаваемых подразделению – отображает наполнение курсов, преподаваемых любыми кафедрами для направлений/специальностей и профилей, подготовка по которым закреплена за указанными подразделениями (как выпускающими кафедрами, или факультетами/институтами на которых ведется подготовка).

В двух последних случаях отображается дерево структуры ПГУ, в котором можно выделить для просмотра одно или несколько подразделений. Далее происходит переход к статистике по уровням образовательных программ (рис. 1), от которой можно перейти к данным по отдельному уровню образования вне зависимости от наполнения курсов либо к курсам этого уровня, отнесенным к одной из категорий, с точки зрения полноты выкладки образовательного контента или с точки зрения использования образовательных форумов в курсе. Данные включают список профилей подготовки, в которых есть хотя бы один курс, соответствующий критерию отбора, а также число таких уровней. После выбора конкретного профиля появляется список удовлетворяющих критерию отбора курсов, упорядоченных по годам обучения (например, только курсы без контента или курсы, где ведется образовательный форум). По каждому курсу отображается название, преподающая кафедра, ответственный за ЭИОС от этой кафедры, преподаватели дисциплины, количество ресурсов по дисциплине и их список.



| Уровень | Число курсов в уровне | Тип курсов | Число курсов в типе |
|------------------------|-----------------------|---|---------------------|
| Всего загружено | 29822 | ПУСТЫЕ (содержит курсы без ресурсов или только с одним любым ресурсом) | 728 |
| | | БЕЗ АННОТАЦИИ ИЛИ РАБОЧЕЙ ПРОГРАММЫ (содержит курсы, в которых в первой или второй темах первый ресурс НЕ файл) | 90 |
| | | ПОЛНОСТЬЮ/ЧАСТИЧНО ЗАПОЛНЕННЫЕ (содержит курсы, не попавшие в первые два раздела) | 29004 |
| | | ФОРУМ НЕ ВЕДЕТСЯ (содержит курсы, форумы которых содержат менее двух сообщений) | 29062 |
| | | ФОРУМ ИСПОЛЬЗУЕТСЯ (содержит курсы, форумы которых содержат два и более сообщений) | 760 |
| Из них | | | |
| Бакалавриат | 18959 | ПУСТЫЕ (содержит курсы без ресурсов или только с одним любым ресурсом) | 374 |
| | | БЕЗ АННОТАЦИИ ИЛИ РАБОЧЕЙ ПРОГРАММЫ (содержит курсы, в которых в первой или второй темах первый ресурс НЕ файл) | 34 |
| | | ПОЛНОСТЬЮ/ЧАСТИЧНО ЗАПОЛНЕННЫЕ (содержит курсы, не попавшие в первые два раздела) | 18551 |
| | | ФОРУМ НЕ ВЕДЕТСЯ (содержит курсы, форумы которых содержат менее двух сообщений) | 18454 |
| | | ФОРУМ ИСПОЛЬЗУЕТСЯ (содержит курсы, форумы которых содержат два и более сообщений) | 505 |
| Магистратура | 3104 | ПУСТЫЕ (содержит курсы без ресурсов или только с одним любым ресурсом) | 127 |
| | | БЕЗ АННОТАЦИИ ИЛИ РАБОЧЕЙ ПРОГРАММЫ (содержит курсы, в которых в первой или второй темах первый ресурс НЕ файл) | 16 |
| | | ПОЛНОСТЬЮ/ЧАСТИЧНО ЗАПОЛНЕННЫЕ (содержит курсы, не попавшие в первые два раздела) | 2961 |

Рис.

Помимо прочего, данный компонент может быть полезен любому пользователю ЭИОС для того, чтобы узнать, кто преподает ту или иную дисциплину на любой специальности, состав методического обеспечения, размещенного в ЭИОС. Но увидеть содержание размещенного на курсе может только имеющий доступ к нему.

Хотя сведения из компонента в любой момент доступны любому пользователю, для стимулирования быстреего заполнения в период первоначального заполнения ЭИОС на основе информации из компонента регулярно формировались рейтинги загрузки по кафедрам и факультетам. В рейтинге для каждого подразделения указывалось общее число преподаваемых курсов, количество курсов, по которым загружен контент, и процент загрузки. Файлы рейтинга регулярно рассылались по кафедрам и факультетам, рейтинг регулярно заслушивался на ректорате. Такой соревновательный эффект, подкрепленный административными стимулами, позволил за 8 месяцев наполнить контентом около 30 000 курсов, загруженных в ЭИОС.

2) Контроль наличия обязательной информации, отображаемой на портале.

На портале вуза должны быть представлены сведения обо всех реализуемых образовательных программах. Так как в ПГУ одновременно реализуются около 600 учебных планов высшего и среднего профессионального образования, в режиме администрирования при просмотре портала выводится надпись «Не загружены обязательные документы» для тех направлений и специальностей подготовки, для которых не размещены учебный план, календарный график или характеристика образовательной программы. Рабочие программы и аннотации всех дисциплин образовательной программы отображаются на портале из ЭИОС [2], и контроль их наличия осуществляется рассмотренным выше компонентом.

Также на портале должна быть представлена информация обо всех преподавателях вуза. В ПГУ сведения о должности, стаже, образовании, ученой степени, звании, повышении квалификации отображаются из информационной системы управления кадров, а о преподаваемых дисциплинах – из ЭИОС. Если в подразделении отсутствует обязательная информация о ком-то из преподавателей, зарегистрированным пользователям отображается предупреждение, что позволяет оперативно исправлять возможные проблемы.

3) Набор онлайн-инструментов в составе ЭИОС, позволяющих осуществлять мониторинг и контроль:

- выставления баллов в контрольных точках (здесь же отображается наличие настройки БРС);
- посещаемости студентов, выставленной преподавателями;
- баллов текущей успеваемости студентов;
- промежуточной успеваемости студентов (с детализацией на баллы, полученные на каждой контрольной точке и непосредственно в ходе аттестации);
- сводных результатов зачетов и экзаменов по группам.

Все указанные инструменты доступны сотрудникам выпускающих кафедр (полезны, в первую очередь, заведующим кафедрами и кураторам групп), а также сотрудникам деканатов. Преподаватели, проводящие занятия, видят всю эту информацию по своим дисциплинам непосредственно в своих электронных журналах.

4) Сводные отчеты по внесению информации в ЭИОС. Формируются в виде xls файлов, включают сведения:

- о настройке БРС и внесении данных о текущей успеваемости студентов;
- о внесении информации о посещаемости студентов;
- о занесении тем курсовых;
- о настройке дисциплин по выбору.

Все отчеты представляют данные по группам и учебным дисциплинам в формате, позволяющем легко выполнять анализ ситуации по любому факультету или кафедре.

Наличие достаточно развитых инструментов мониторинга работы с ЭИОС позволяет достаточно оперативно обнаруживать проблемные ситуации и принимать необходимые организационные решения. Своевременное и полное внесение информации в ЭИОС позволит повысить оперативность и качество управленческих решений, снизить объем бумажных документов, обеспечить единство и достоверность информации.

Библиографический список

1. Мещеряков, В. А. Электронная информационно-образовательная среда университета в эру ФГОС ВО / В. А. Мещеряков, А. В. Антонов, А. М. Бершадский, И. Г. Кревский // Научно-образовательная информационная среда XXI века : материалы X Всерос. науч.-практ. конф. (20–23 сентября 2016 г.) – Петрозаводск, 2016. – С. 108–111.
2. Кревский, И. Г. ЭИОС вуза как ключевой элемент его информатизации / И. Г. Кревский, А. В. Антонов // Новые информационные технологии и системы : сб. науч. ст. XIV Междунар. науч.-техн. конф., посвящ. 70-летию кафедры «Вычислительная техника» и 30-летию кафедры «Системы автоматизированного проектирования» (г. Пенза, 22–24 ноября 2017 г.). – Пенза : Изд-во ПГУ, 2017. – С. 358–360.

В. В. Мелёшкин

Пензенский государственный университет, Пенза, Россия

КОНЦЕПЦИЯ АВТОМАТИЗАЦИИ СОЗДАНИЯ ЦВЕТОВОГО ДИЗАЙНА САЙТА

Рассматриваются процедура создания цветового дизайна веб-сайта, правила составления цветовой схемы и процедура вёрстки шаблона системы управления контентом (CMS). Рассмотрены существующие генераторы шаблонов для CMS Wordpress, выявлены их достоинства и недостатки. На основе исследования обоснована необходимость создания генератора шаблонов, в котором будут устранены недостатки существующих аналогов и добавлена функция автоматического создания цветового дизайна веб-сайта.

Вследствие роста сети Интернет появилась новая область информационных технологий – веб-дизайн, основной задачей которой является проектирование интерфейсов для сайтов и web-приложений.

При проектировании дизайна веб-сайта основной задачей дизайнера является создание макета (mock-up) средствами многофункционального графического редактора, например Adobe Photoshop либо Sketch (доступен на Mac OS). В макете дизайнер должен отрисовать основные страницы будущего сайта. На данном этапе работы дизайнера ключевым моментом является подбор цветов. Правильно составленная цветовая схема делает сайт эстетически приятным, а ошибки на данном этапе способны испортить впечатление. Как правило, дизайнер выбирает один основной цвет и уже под него подбирает палитру из 2–3 цветов, сочетающихся с основным.

Подборка цветовой схемы может осуществляться в соответствии с корпоративными цветами, если сайт делается для компании. Цвет также может зависеть от тематики сайта, например, если он посвящён садовому декору, основным цветом для него станет зелёный.

Иногда цвета подбираются в соответствии с желанием заказчика. В любом случае для облегчения данного процесса созданы специальные программы, например: Adobe Color CC [1], Color Scheme [2], Color Rotate [3].

Работают подобные приложения на основе принципа подбора цветов по цветовому кругу.

В данном формате цветовой спектр представлен в виде круга, разделенного на сектора разных цветов. В подобном приложении дизайнеру нужно лишь указать код основного цвета, после чего программа автоматически подберёт наиболее сочетающиеся с ним дополнительные цвета по одной из следующих схем.

Аналоговая триада – цветовую схему образуют цвета, расположенные по соседству слева и права от основного цвета.

Комплементарная цветовая схема – состоит из цветов, расположенных на цветовом круге друг напротив друга.

Контрастная триада – это вариант комплементарного сочетания с той лишь разницей, что вместо цвета, противоположного основному, используются цвета, соседние с противоположным.

Классическая триада – данную цветовую схему образуют три равноудаленных друг от друга цвета.

Существуют и другие варианты цветовых схем, но в основе каждой из них лежат перечисленные выше.

После того как дизайнер распределит полученную цветовую схему по макету и выполнит прочие действия, необходимые для завершения работы над ним, макет передают программисту для его последующей вёрстки.

Скорее всего, программист не будет реализовывать полноценный сайт с нуля, так как существует множество доступных систем управления контентом (Content Management System), которые предоставляют пользователям функции создания, редактирования, контроля и организации веб-страниц. Вот три наиболее распространённых в России CMS: WordPress [4], Joomla [5], Drupal [6]. Дальнейшая вёрстка макета, созданного дизайнером, представляет собой создание шаблона для одной из CMS.

Недостатком данной процедуры является, необходимость обладать знанием веб-дизайна и веб-программирования, что для большинства пользователей делает создание собственного шаблона невозможным. Решением данной проблемы являются генераторы шаблонов, которые позволяют пользователю настроить шаблон под себя. Но подобных сервисов в настоящий момент существует немного: Lubith.com [7], Yvoschaap.com [8], Wpthemegenerator.com [9], да и те имеют ряд недостатков:

- непривлекательные и устаревшие стандартные шаблоны;
- всего один исходный шаблон;
- отсутствие русскоязычной версии;
- Lubith.com и Wpthemegenerator.com являются платными.

В качестве выхода из сложившейся ситуации можно предложить концепцию генератора шаблонов, который будет учитывать ошибки предыдущих систем. Работать он должен по следующему принципу:

- 1) для выбора пользователю будут предоставлены несколько различных шаблонов;
- 2) пользователь сможет изменять параметры шаблона;
- 3) внесённые пользователем настройки будут сохранены в суперглобальный массив POST, который создаётся при запуске языка программирования PHP и позволяет передавать данные в другие файлы. Именно этим методом полученные от пользователя настройки будут переданы в файлы шаблона, после чего в них произойдут изменения;
- 4) затем пользователь сможет скачать индивидуальный шаблон.

А для того, чтобы помочь пользователю создать правильный дизайн, в данную систему должна быть встроена функция автоматического подбора цветов для шаблона на основе различных цветовых схем. Для этого ему нужно только ввести основной цвет в формате RGB (цвет кодируется с помощью трёх основных цветов, интенсивности которых задаются в диапазоне от 0 до 255) или шестнадцатеричной модели цветов HTML, выбрать нужную цветовую схему, после чего система вычислит дополнительные цвета и распределит автоматически их по шаблону.

Определение цвета будет происходить по цветовой модели HSV, в которую будут конвертированы значения начального цвета, полученные от пользователя. Цвет в модели HSV формируется из трех параметров:

- цветовой тон (hue) – принимает значение от 0 до 360° и отвечает за то, какой цвет спектра будет выбран;
- насыщенность (saturation) – принимается значение от 0 до 100. Чем выше данный параметр, тем «чище» цвет;
- яркость (value) – варьируется в пределах от 0 до 100. Чем больше данный параметр, тем ярче цвет.

Перевод из RGB в HSV осуществляется по следующим формулам [10]:

$$H = \begin{cases} 0, & \text{если } \Delta = 0, \\ 60 \times \frac{G' - B'}{\Delta} \bmod 6, & \text{если } MAX = R', \\ 60 \times \frac{B' - R'}{\Delta} + 2, & \text{если } MAX = G', \\ 60 \times \frac{R' - G'}{\Delta} + 4, & \text{если } MAX = B'. \end{cases}$$

$$S = \begin{cases} 0, & \text{если } MAX = 0, \\ \frac{\Delta}{MAX}, & \text{если } MAX \neq 0. \end{cases}$$

$$V = MAX,$$

где R', G', B' – интенсивности цветов, приведенные к единичному диапазону, $R' = R / 255$, $G' = G / 255$, $B' = B / 255$; MAX – максимальное значение из R', G', B' ; MIN – минимальное значение из R', G', B' ; Δ – разность ($MAX - MIN$).

Цветовую модель HSV можно графически представить в виде цилиндра (рис. 1), в котором Hue – это полярный угол, Saturation – радиус-вектор, а Value – это высота.



Рис. 1. Наглядное представление модели HSV

Цветовая модель HSV идеально подходит для нашей задачи, потому что для получения нужной цветовой схемы достаточно изменить значение только одного параметра Hue. Например, комплементарный (контрастный) цвет H' можно получить по следующей формуле:

$$H' = \begin{cases} H - 180, & \text{если } H \geq 180, \\ H + 180, & \text{если } H < 180. \end{cases}$$

Затем полученный цвет нужно будет перевести обратно в цветовую модель RGB. Делается это по следующим формулам:

$$C = (V' - S'),$$

$$X = C \cdot \left(1 - \left| \left(\frac{H}{60} \right) \bmod 2 - 1 \right| \right),$$

$$(R', G', B') = \begin{cases} (C, X, 0), & \text{если } H \in [0, 60], \\ (X, C, 0), & \text{если } H \in [60, 120], \\ (0, C, X), & \text{если } H \in [120, 180], \\ (0, X, C), & \text{если } H \in [180, 240], \\ (X, 0, C), & \text{если } H \in [240, 300], \\ (C, 0, X), & \text{если } H \in [300, 360], \end{cases}$$

$$\begin{aligned}M &= V' - C', \\R &= ((R' + M) \cdot 255), \\G &= ((G' + M) \cdot 255), \\B &= ((B' + M) \cdot 255),\end{aligned}$$

где $H \in [0, 360]$, а S', V' – значения насыщенности и яркости, приведенные к единичному диапазону:

$$\begin{aligned}S' &= S/100, \\V' &= V/100.\end{aligned}$$

Таким образом, в результате реализации данной концепции будет разработана система автоматического создания цветового дизайна веб-сайта, лишенная недостатков существующих аналогов, а функция автоматического подбора цветов позволит пользователям избежать ошибок в дизайне.

Библиографический список

1. Приложение для создания цветовых схем Adobe Color CC. – URL: <https://color.adobe.com>
2. Приложение для создания цветовых схем Color Scheme. – URL: <https://colorscheme.ru/>
3. Приложение для создания цветовых схем Color Rotate. – URL: <http://web.colorotate.org/>
4. Система создания и управления сайтом WordPress. – URL: <https://ru.wordpress.com/>
5. Система создания и управления сайтом WordPress Joomla. – URL: <https://www.joomla.org/>
6. Система создания и управления сайтом WordPress Drupal. – URL: <https://www.drupal.org/>
7. Генератор тем для CMS Wordpress Lubith. – URL: <http://www.lubith.com/>
8. Генератор тем для CMS Wordpress Yvoschaap. – URL: <http://www.yvoschaap.com/wpthe-megen/>
9. Генератор тем для CMS Wordpress WP Theme Generator. – URL: <http://www.wpthemegenerator.com/>
10. HSV (цветовая модель) : свободная энциклопедия. – URL: https://ru.wikipedia.org/HSV_цветовая_модель (дата обновления: 23.05.2018).

СОДЕРЖАНИЕ

АППАРАТНО-ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СИСТЕМ УПРАВЛЕНИЯ

| | |
|--|----|
| А. В. Чернышов ПРОБЛЕМА ПРОЕКТИРОВАНИЯ АРХИВНЫХ ХРАНИЛИЩ ЭЛЕКТРОННОЙ ИНФОРМАЦИИ | 3 |
| В. В. Бурлин, Л. В. Гурьянов О РЕАЛИЗАЦИИ АЛГОРИТМА БЕРНШТЕЙНА–ВАЗИРАНИ НА ЯЗЫКЕ PUTHON | 6 |
| Р. Н. Федюнин АРИФМЕТИКО-ЛОГИЧЕСКИЕ УСТРОЙСТВА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ, ДЛЯ ОБРАБОТКИ ОДНОРОДНЫХ МАТРИЦ | 8 |
| В. Н. Дубинин, А. С. Войнов, И. В. Сенокосов ПРЕДСТАВЛЕНИЯ 3D-КОНСТРУКЦИЙ В LEGO НА ЯЗЫКЕ ПРОЛОГ И ИХ ПРЕОБРАЗОВАНИЯ | 11 |
| К. Н. Савинов, Д. А. Данилин, Р. А. Торопкин АНАЛИЗ СОВРЕМЕННЫХ РЕШЕНИЙ ДЛЯ РАЗРАБОТКИ КРОСС-ПЛАТФОРМЕННЫХ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ | 15 |
| М. А. Барышев, Л. В. Гурьянов ПРИМЕНЕНИЕ КЛАСТЕРНОГО АНАЛИЗА ДЛЯ РЕШЕНИЯ ЗАДАЧ МАРКЕТИНГА НА ПРЕДПРИЯТИИ | 17 |
| И. А. Долгова, И. П. Грибов ПЕРСПЕКТИВЫ ИСПОЛЬЗОВАНИЯ ТЕХНОЛОГИИ БЛОКЧЕЙН В ЭКОНОМИКЕ | 19 |
| И. А. Долгова, И. С. Золина ИСПОЛЬЗОВАНИЕ БАГ-ТРЕКИНГОВЫХ СИСТЕМ ДЛЯ ТЕСТИРОВАНИЯ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ | 22 |
| О. С. Дорофеева, Б. В. Казаков, И. А. Казакова ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА К РАЗРАБОТКЕ И МОДЕЛИРОВАНИЮ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ | 25 |
| О. С. Дорофеева, И. А. Казакова, Б. В. Казаков ПРОГРАММНЫЕ СРЕДСТВА РАСПОЗНАВАНИЯ ЯЗЫКОВИ ГРАММАТИК | 27 |
| Ю. Н. Косников, Т. Х. Хоанг ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ ИЗОБРАЗИТЕЛЬНЫХ И ТОЧНОСТНЫХ СВОЙСТВ СМЕШИВАЮЩИХ ФУНКЦИЙ В ЗАДАЧАХ ВИЗУАЛИЗАЦИИ ПРОСТРАНСТВЕННЫХ ОБЪЕКТОВ | 29 |
| К. Б. Новиков, А. Ю. Боровиков РАССМОТРЕНИЕ ВОЗМОЖНОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ МОДУЛЯ ДОВЕРЕННОЙ ЗАГРУЗКИ ОПЕРАЦИОННОЙ СИСТЕМЫ | 33 |
| Е. Н. Прошкина ОНТОЛОГИЧЕСКИЙ ИНЖИНИРИНГ В ПРОЕКТИРОВАНИИ ИНФОРМАЦИОННЫХ СИСТЕМ | 36 |
| Д. А. Родионов АНАЛИЗ ВОЗМОЖНОСТЕЙ МИКРОСХЕМ ФИРМЫ FTDI | 40 |
| Д. С. Смирнов РЕАЛИЗАЦИЯ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ВЕБ-САЙТОВ НА ОСНОВЕ ВЕРОЯТНОСТНОГО АВТОМАТА | 42 |
| Д. А. Хоцян, О. Э. Яремко МЕТОД ФУНДАМЕНТАЛЬНЫХ РЕШЕНИЙ В ТЕОРИИ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ ЭЛЛИПТИЧЕСКОГО ТИПА | 45 |

| | |
|--|----|
| Л. В. Тельнова, Н. Ю. Кудряшова МЕТОД КВАЗИОБРАЩЕНИЯ ДЛЯ РЕШЕНИЯ СТАЦИОНАРНОЙ ГРАНИЧНОЙ ЗАДАЧИ ТЕПЛОПРОВОДНОСТИ | 48 |
| Б. А. Кирьяк, Ю. Ю. Горюнов РАЗРАБОТКА ПРОГРАММ ОБЕСПЕЧЕНИЯ МОДЕЛИРОВАНИЯ СИСТЕМ СЕТЯМИ ПЕТРИ..... | 52 |
| Г. В. Бобрышева, А. О. Звозникова ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ..... | 54 |
| НА ПРЕДПРИЯТИИ | 54 |
| А. В. Травянко, Г. В. Бобрышева, Г. О. Звозникова ПРОГРАММНЫЕ РЕШЕНИЯ ДЛЯ ОЦЕНКИ РИСКОВ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ | 57 |

ИНТЕЛЕКТУАЛЬНЫЕ СИСТЕМЫ И МАШИННОЕ ОБУЧЕНИЕ

| | |
|---|----|
| И. Ю. Шполянская, А. Р. Загиров МЕТОДЫ МАШИННОГО ОБУЧЕНИЯ В СОЗДАНИИ РЕКОМЕНДАТЕЛЬНЫХ СИСТЕМ..... | 61 |
| Б. С. Ахметов, Л. М. Кыдыралина МОДЕЛЬ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОДДЕРЖКИ РЕШЕНИЯ ПО ФИНАНСОВЫМ ВЛОЖЕНИЯМ В КИБЕРБЕЗОПАСНОСТЬ ИНФОРМАЦИОННО-ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ УНИВЕРСИТЕТА..... | 65 |
| Б. С. Башев, П. П. Макарычев МЕТОД ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НИЗКОГО РАЗРЕШЕНИЯ В СИСТЕМАХ РАСПОЗНАВАНИЯ ТЕКСТА..... | 68 |
| П. П. Макарычев МОДЕЛЬДАТЧИКА РАССОГЛАСОВАНИЙ ДИРЕКЦИОННЫХ УГЛОВ В СЛЕДЯЩЕЙ СИСТЕМЕ | 71 |
| М. М. Алкезуини, В. И. Горбаченко РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ НА СЕТЯХ РАДИАЛЬНЫХ БАЗИСНЫХ ФУНКЦИЙ..... | 74 |
| К. Н. Савинов, Д. А. Данилин ИСПОЛЬЗОВАНИЕ МАТЕМАТИЧЕСКИХ ПАКЕТОВ ПЛАТФОРМЫ .NETFRAMEWORK ДЛЯ СОЗДАНИЯ СЕТИ НЕЙРОНОВ «КРАМЕРА–ФОН МИЗЕСА»..... | 79 |
| Д. В. Спирин АЛГОРИТМ АВТОМАТИЗИРОВАННОГО ОБУЧЕНИЯ ДИАЛОГОВЫХ СИСТЕМ | 81 |
| А. Ю. Тычков, А. К. Алимуратов, П. П. Чураков ОБНАРУЖЕНИЕ МАРКЕРОВ РЕЧЕВЫХ СИГНАЛОВ С ИСПОЛЬЗОВАНИЕМ ТЕОРИИ ГИЛЬБЕРТА–ХУАНГА | 84 |
| А. П. Писарев, М. А. Писарев, Е. М. Голобокова АКТУАЛЬНОСТЬ РАЗРАБОТКИ ПРОГРАММНЫХ И АЛГОРИТМИЧЕСКИХ СРЕДСТВ ДЛЯ СОЗДАНИЯ ПРИБОРОВ ПРОГНОЗИРОВАНИЯ ИНФАРКТА МИОКАРДА | 89 |
| А. П. Писарев, М. А. Писарев, Е. М. Голобокова ИСПОЛЬЗОВАНИЕ АДАПТИВНЫХ И НЕЛИНЕЙНЫХ ФИЛЬТРОВ ПРИ ПОЛУЧЕНИИ И ОБРАБОТКЕ КОМПОНЕНТ ПУЛЬСОВОЙ ВОЛНЫ | 91 |
| Г. С. Тупицин, А. И. Топников УСОВЕРШЕНСТВОВАННЫЙ НЕЙРОСЕТЕВОЙ АЛГОРИТМ ПОДАВЛЕНИЯ ШУМА В РЕЧЕВЫХ СИГНАЛАХ НА ОСНОВЕ МЯГКОЙ МАСКИ..... | 96 |

| | |
|--|-----|
| А. К. Алимуратов, А. Ю. Тычков, П. П. Чураков СПОСОБ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ОЦЕНКИ ПСИХОЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ОПЕРАТОРОВ СИСТЕМ УПРАВЛЕНИЯ С ВЫСОКОЙ СТЕПЕНЬЮ ОТВЕТСТВЕННОСТИ | 99 |
| С. Е. Вятчанин, Е. А. Малыгина, С. В. Туреев ОСОБЕННОСТИ АВТОМАТИЧЕСКОГО ОБУЧЕНИЯ СЕТИ НЕЙРОНОВ «КРАМЕРА–ФОН МИЗЕСА» НА МАЛЫХ ВЫБОРКАХ | 102 |
| А. И. Иванов, Е. А. Малыгина, А. И. Солопов ИСПОЛЬЗОВАНИЕ НЕЙРОНОВ КВАДРАТА СРЕДНЕГО ГЕОМЕТРИЧЕСКОГО ПЛОТНОСТЕЙ РАСПРЕДЕЛЕНИЯ ЗНАЧЕНИЙ ВЕРОЯТНОСТЕЙ ПРИ МНОГОМЕРНОМ АНАЛИЗЕ ДАННЫХ | 104 |
| А. С. Башев МЕТОД ДИНАМИЧЕСКОГО ИЗМЕНЕНИЯ ПАРАМЕТРА БДИТЕЛЬНОСТИ В НЕЙРОННОЙ СЕТИ ART-1 | 107 |
| О. Н. Бодин, Г. Ф. Убиенных, А. Г. Убиенных МОДЕЛИРОВАНИЕ ВИХРЕВЫХ АВТОВОЛН ТИПА РИЕНТРИ В ПРЕДСЕРДИЯХ ЧЕЛОВЕКА | 110 |
| А. С. Никонов, И. П. Бурукина ПРИМЕНЕНИЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЯ ДЛЯ РАСПОЗНАВАНИЯ РЕЧИ..... | 114 |
| А. А. Валько, А. Ф. Валько, А. С. Ивлиев ПРИМЕНЕНИЕ ВОЗМОЖНОСТЕЙ СОВРЕМЕННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОРРЕКТИРОВКИ ВЛИЯНИЯ ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ НА ПОЛЬЗОВАТЕЛЯ | 116 |
| М. А. Вахромеев, А. Д. Глебов НЕЙРОННЫЕ СЕТИ С ИЗМЕНЯЕМОЙ ТОПОЛОГИЕЙ И ОСОБЕННОСТИ ИХ ОБУЧЕНИЯ..... | 119 |
| Д. А. Голдуева, М. А. Мокшанина АНАЛИЗ ПОВЕРХНОСТИ ЗЕМЛИ С ПОЗИЦИИ СТОХАСТИЧЕСКОЙ ГЕОМЕТРИИ И ФУНКЦИОНАЛЬНОГО АНАЛИЗА | 121 |
| Е. С. Серов, Л. В. Гурьянов МЕТОД РЕШЕНИЯ ЗАДАЧ СПОРТИВНОГО ПРОГРАММИРОВАНИЯ НА ОСНОВЕ ПРЕФИКСНОГО ДЕРЕВА | 124 |
| Ю. И. Евсеева, А. С. Бождай, А. М. Бершадский КОНЦЕПЦИЯ РЕФЛЕКСИИ И АРХИТЕКТУРА РЕФЛЕКСИВНОЙ ПРОГРАММНОЙ СИСТЕМЫ | 126 |
| Ю. И. Евсеева, А. А. Гудков ИСПОЛЬЗОВАНИЕ КОНЦЕПЦИИ ИНЖЕНЕРИИ ЛИНЕЕК ПРОГРАММНЫХ ПРОДУКТОВ ДЛЯ СОЗДАНИЯ АДАПТИВНЫХ ПРОГРАММНЫХ СИСТЕМ..... | 129 |
| Е. А. Кольчугина ТЕОРИЯ НУМЕРАЦИЙ КАК ОСНОВА ДЛЯ ПОСТРОЕНИЯ САМООРГАНИЗУЮЩИХСЯ ПРОГРАММНЫХ СИСТЕМ | 133 |
| А. Ю. Тимонин ЛЕКСИЧЕСКИЙ АНАЛИЗ ИСХОДНОГО НЕСТРУКТУРИРОВАННОГО ТЕКСТА В ПРОЦЕССЕ СОЦИАЛЬНОГО ПРОФИЛИРОВАНИЯ..... | 135 |
| А. Е. Хохлов МОДЕЛИРОВАНИЕ ВЫНУЖДЕННЫХ КОЛЕБАНИЙ ПЛАСТИН НА КЛЕТОЧНЫХ НЕЙРОННЫХ СЕТЯХ | 139 |
| С. В. Тумасов РАСПОЗНАВАНИЕ РЕЧИ В СЕТЯХ ПЕРЕДАЧИ ДАННЫХ | 142 |

УПРАВЛЕНИЕ В СОЦИАЛЬНЫХ, ЭКОНОМИЧЕСКИХ И ТЕХНИЧЕСКИХ СИСТЕМАХ

| | |
|--|-----|
| <i>О. О. Ашкирина, З. И. Баусова, М. И. Гринцов, Н. В. Дорогова</i> ИСПОЛЬЗОВАНИЕ БИНАРНОЙ ЛОГИСТИЧЕСКОЙ МОДЕЛИ В МЕДИЦИНСКИХ ИССЛЕДОВАНИЯХ | 145 |
| <i>А. К. Горелова</i> АНАЛИЗ ВОЗМОЖНОСТЕЙ МОБИЛЬНЫХ СИСТЕМ ЭЛЕКТРОННОГО ЗДРАВООХРАНЕНИЯ | 148 |
| <i>З. И. Баусова, А. Ю. Старикова, А. Н. Гамазина, Ю. Н. Дугина</i> АНАЛИЗ МЕТОДОВ РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ..... | 151 |
| <i>А. И. Жиганов, Н. П. Елисейев, Л. В. Гурьянов</i> СОВРЕМЕННЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ МАСШТАБИРОВАНИЯ И УПРАВЛЕНИЯ КОНТЕЙНЕРИЗИРОВАННЫМИ ПРИЛОЖЕНИЯМИ ДЛЯ РАСПРЕДЕЛЕННЫХ КЛАСТЕРНЫХ СИСТЕМ | 154 |
| <i>Г. Ю. Жулимов, А. Д. Мальцева, З. И. Баусова, К. М. Буданов</i> РЕШЕНИЕ ЭКОНОМИЧЕСКИХ ЗАДАЧ, СВЯЗАННЫХ С КОНФЛИКТНЫМИ СИТУАЦИЯМИ..... | 156 |
| <i>Н. В. Сорокина, И. Ю. Балашова</i> РАЗРАБОТКА АППАРАТНО-ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ УПРАВЛЕНИЯ ЭЛЕКТРООБОГРЕВОМ ТЕХНОЛОГИЧЕСКОГО ОБОРУДОВАНИЯ АЭС | 159 |
| <i>И. Ю. Балашова</i> ИНТЕЛЛЕКТУАЛИЗАЦИЯ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО УПРАВЛЕНИЮ ПЕРСОНАЛОМ ОРГАНИЗАЦИИ НА ОСНОВЕ НЕЧЕТКО-ОНТОЛОГИЧЕСКОЙ МОДЕЛИ | 162 |
| <i>Е. С. Ефимова, Я. А. Колесникова</i> САМЫЕ РАСПРОСТРАНЕННЫЕ ERP-СИСТЕМЫ В РОССИЙСКОЙ ФЕДЕРАЦИИ..... | 165 |
| <i>Л. Р. Фионова, М. А. Катышева</i> ДОКУМЕНТИРОВАНИЕ ЭЛЕКТРОННЫХ ТОРГОВ ДЛЯ МЕДИЦИНСКИХ УЧРЕЖДЕНИЙ | 167 |
| <i>Л. Р. Фионова</i> УПРАВЛЕНИЕ ЭЛЕКТРОННЫМ ДОКУМЕНТООБОРОТОМ В ОРГАНАХ ВЛАСТИ | 170 |

СЕТИ ЭВМ И ТЕЛЕКОММУНИКАЦИИ

| | |
|--|-----|
| <i>А. А. Корченко, Н. К. Жумангалиева</i> СТРУКТУРНАЯ МОДЕЛЬ СИСТЕМЫ ВЫЯВЛЕНИЯ ВТОРЖЕНИЙ..... | 174 |
| <i>В. И. Волчихин, С. А. Зинкин, Н. С. Карамышева</i> ОРГАНИЗАЦИЯ МЕТАКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА ПЛАТФОРМЕ АВТОНОМНЫХ МОБИЛЬНЫХ АГЕНТОВ | 176 |
| <i>Мустафа СадекДжафар, С. А. Зинкин</i> О РАЗВИТИИ КОНЦЕПЦИИ «СЕТЬ – ЭТО КОМПЬЮТЕР»..... | 186 |
| <i>А. А. Яужев</i> ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ВСТРАИВАЕМЫХ СИСТЕМ НА ОСНОВЕ МИКРОКОНТРОЛЛЕРОВ | 193 |
| <i>А. О. Семенов</i> О РАСПРЕДЕЛЕНИИ ПОЛОСЫ ПРОПУСКАНИЯ В РАМКАХ КОММУТАТОРА НА БАЗЕ СТОХАСТИЧЕСКОЙ ДИСПЕТЧЕРИЗАЦИИ | 195 |

| | |
|---|-----|
| Д. А. Замятин, В. В. Кольга ОСНОВНЫЕ АСПЕКТЫ ПРОЕКТИРОВАНИЯ ОРБИТАЛЬНОГО ЦЕНТРА ХРАНЕНИЯ И ПЕРЕДАЧИ ДАННЫХ | 198 |
| К. И. Никишин, Е. И. Гурин, Н. Н. Коннов РЕАЛИЗАЦИЯ SCHEDULE TIME-TRIGGERED ETHERNET С ИСПОЛЬЗОВАНИЕМ ПЛИС | 200 |
| И. Ю. Фролов ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ СИНХРОННОГО И АСИНХРОННОГО СБРОСА | 203 |
| К. Ю. Шабалова ПРОБЛЕМА ОРИЕНТАЦИИ МОБИЛЬНЫХ УСТРОЙСТВ В UNITY 3D..... | 205 |

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

| | |
|---|-----|
| А. А. Турмышов, Н. Ю. Митрохина ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ЗАДАЧ В КУРСЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА» | 207 |
| А. П. Писарев, Е. М. Голобокова, Ю. В. Терякова НОВЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ | 210 |
| Т. В. Глотова, П. В. Ечная БАЗА ДАННЫХ ПРОГРАММНЫХ СРЕДСТВ, ИСПОЛЬЗУЕМЫХ В ОБУЧЕНИИ..... | 214 |
| Т. В. Глотова, О. В. Красильникова КОМПОНЕНТЫ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ | 216 |
| Л. В. Гурьянов, Е. А. Дзюба, Д. А. Миронов РАЗРАБОТКА СТУДЕНЧЕСКИХ КУРСОВ НА ПЛАТФОРМЕ SOFTWARE-ACADEMY | 219 |
| А. Г. Мирзалиев WEB-ОРИЕНТИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ ОЦЕНКИ ЭФФЕКТИВНОСТИ РАБОТЫ ПРЕПОДАВАТЕЛЕЙ ВУЗА | 221 |
| С. В. Самуйлов, П. М. Фрунзе МЕТОДИКА АНАЛИЗА АЛГОРИТМОВ ОБРАБОТКИ ДАННЫХ..... | 224 |
| А. Е. Хохлов, Ю. Н. Дугина ОБЕСПЕЧЕНИЕ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ ТЕХНОЛОГИЯМИ «1С» ПРИ ПОДГОТОВКЕ БАКАЛАВРОВ ПО ПРОФИЛЮ «ПРИКЛАДНАЯ ИНФОРМАТИКА В ЭКОНОМИКЕ» | 226 |
| А. В. Ширканов ПЕРВЫЕ РЕЗУЛЬТАТЫ ИСПОЛЬЗОВАНИЯ КОМПОНЕНТА КОНТРОЛЯ НАПОЛНЕНИЯ УЧЕБНЫХ КУРСОВ ЭИОС | 228 |
| И. Г. Кревский, А. В. Антонов, М. В. Деев, Э. А. Зубриядина, И. А. Рензяева, А. В. Ширканов УПРАВЛЕНИЕ ВНЕДРЕНИЕМ ИНФОРМАЦИОННОЙ СИСТЕМЫ ВУЗА..... | 228 |
| В. В. Мелёшкин КОНЦЕПЦИЯ АВТОМАТИЗАЦИИ СОЗДАНИЯ ЦВЕТОВОГО ДИЗАЙНА САЙТА..... | 232 |

Научное издание

**НОВЫЕ ИНФОРМАЦИОННЫЕ
ТЕХНОЛОГИИ И СИСТЕМЫ**

СБОРНИК НАУЧНЫХ СТАТЕЙ

XV Международной научно-технической
конференции,
посвященной 75-летию Пензенского государственного университета

г. Пенза, 20–21 ноября 2018 г.

Материалы печатаются в авторской редакции.

Компьютерная верстка *М. Б. Жучковой*
Дизайн обложки *А. А. Стаценко*

Подписано в печать 00.10.2018.
Формат 60×84¹/₈. Усл. печ. л. 28,13.
Тираж 30. Заказ № .

Издательство ПГУ.
440026, Пенза, Красная, 40.
Тел./факс: (8412) 56-47-33; e-mail: iic@pnzgu.ru