

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
МЕЖДУНАРОДНАЯ АКАДЕМИЯ ИНФОРМАТИЗАЦИИ
СОЮЗ МАШИНОСТРОИТЕЛЕЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ, ИННОВАЦИОННЫХ
И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ РЯЗАНСКОЙ ОБЛАСТИ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

СТНО-2018

**МЕЖДУНАРОДНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ
ФОРУМ**

Сборник трудов

Том 4

Рязань
Book Jet
2018

УДК 004 + 001.1 + 681.2+ 681.2+ 681.3+681.5
С 568

Современные технологии в науке и образовании – СТНО-2018 [текст]: сб. тр. междунар. науч.-техн. форума: в 10 т. Т.4./ под общ. ред. О.В. Миловзорова. – Рязань: Рязан. гос. радиотехн. ун-т, 2018; Рязань. –237 с.,: ил.

Сборник включает труды участников Международного научно-технического форума «Современные технологии в науке и образовании» СТНО-2018.

В сборнике освещаются вопросы математического моделирования, новых технологий в радиотехнике, телекоммуникациях, электротехнике и радиоэлектронике, вопросы полупроводниковой наноэлектроники, приборостроения, лазерной, микроволновой техники, силовой промышленной электроники, новые технологии в измерительной технике и системах, биомедицинских системах, алгоритмическое и программное обеспечение вычислительной техники, вычислительных сетей и комплексов, вопросы систем автоматизированного проектирования, обработки изображений и управления в технических системах, перспективные технологии в машиностроительном и нефтехимическом производствах, новые технологии и методики в высшем образовании, в т.ч. вопросы гуманитарной и физико-математической подготовки студентов, обучения их иностранным языкам, перспективные технологии электронного обучения, в том числе, дистанционного, вопросы экономики, управления предприятиями и персоналом, менеджмента, а также вопросы гуманитарной сферы.

Авторская позиция и стилистические особенности сохранены.

УДК 004 + 001.1 + 681.2+ 681.2+ 681.3+681.5

ISBN 978-5-7722-0293-7

© Рязанский государственный
радиотехнический университет, 2018
© Издательство «Book Jet»,
макет, 2018

ИНФОРМАЦИЯ О МЕЖДУНАРОДНОМ ФОРУМЕ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ» СТНО-2018

Международный научно-технический форум «Современные технологии в науке и образовании» СТНО-2018 состоялся 28.02.2018-02.03.2018 в г. Рязань в Рязанском государственном радиотехническом университете.

В рамках форума «Современные технологии в науке и образовании» СТНО-2018 состоялись четыре Международные научно-технические конференции:

«Современные технологии в науке и образовании. Радиотехника и электроника», секции

- Радиотехнические системы и устройства;
- Телекоммуникационные системы и устройства;
- Цифровые информационные технологии реального времени;
- Промышленная силовая электроника, электроэнергетика и электроснабжение;
- Физика полупроводников, микро- и нанoeлектроника;
- Микроволновая, оптическая и квантовая электроника;
- Современные методы обработки данных;
- Актуальные задачи химических технологий;

«Современные технологии в науке и образовании. Вычислительная техника и автоматизированные системы», секции

- Алгоритмическое и программное обеспечение вычислительных систем и сетей;
- ЭВМ и системы;
- Системы автоматизированного проектирования;
- Информационные системы и защита информации;
- Математические методы в научных исследованиях;
- Обработка изображений и управление в технических системах;
- Геоинформационные и космические технологии;
- Автоматизация производственно-технологических процессов в приборо- и машиностроении;

• Информационно-измерительные устройства и системы в технике и медицине;

«Современные технологии в науке и образовании. Экономика и управление», секции;

• Проблемы рынка: экономика и управление;

• Актуальные проблемы государственного, муниципального и корпоративного управления;

- Менеджмент и организация производства;
- Бухгалтерский учет, анализ и аудит;
- Управление персоналом;
- Экономическая безопасность;

«Современные технологии в науке и образовании. Новые технологии и методы в высшем образовании», секции

- Современные технологии электронного обучения;
- Иностранный язык в техническом вузе;
- Лингвистика и межкультурная коммуникация;
- Направления и формы гуманитаризации высшего образования;
- Методы преподавания и организация учебного процесса в вузе;
- Гуманитарная подготовка студентов;
- Физико-математическая подготовка студентов;
- Технологии обучения и воспитания на военной кафедре.

Организационный комитет Форума:

Чиркин М.В., И.О. ректора, д.ф.-м.н., проф. – председатель

Гусев С.И., проректор по научной работе, д.т.н., проф. – зам. председателя;

Бухенский К.В., проректор по учебной работе, к.ф.-м.н., доц. – зам. председателя;

Миловзоров О.В., зам. директора института магистратуры и аспирантуры, к.т.н, доц. – координатор;

Устинова Л.С., начальник отдела информационного обеспечения – отв. за информационную поддержку;

Трубицына С.Г., вед. инженер – секретарь оргкомитета;

Благодарова И.А., ведущий программист – секретарь оргкомитета;

члены оргкомитета:

Бабаян П.В., к.т.н., доц., заведующий кафедрой «Автоматика и информационные технологии в управлении»;

Витязев В.В., д.т.н., проф., заведующий кафедрой «Телекоммуникации и основы радиотехники»;

Евдокимова Е.Н., д.э.н., проф., декан Инженерно-экономического факультета;

Еремеев В.В., д.т.н., проф., директор НИИ «Фотон»;

Есенина Н.Е., к.п.н., доц., заведующая кафедрой Иностранных языков;

Жулев В.И., д.т.н., проф., заведующий кафедрой "Информационно-измерительная и биомедицинская техника";

Иваненко Р.В., полковник, начальник Военной кафедры;

Кириллов С.Н., д.т.н., проф., заведующий кафедрой "Радиоуправление и связь";

Клейносова Н.П., к.п.н., доц., директор Центра дистанционного обучения;

Ключко В.К., д.т.н., профессор кафедры «Автоматика и информационные технологии в управлении»;

Коваленко В.В., к.т.н., доц., заведующий кафедрой «Химическая технология»;

Корячко В.П., д.т.н., проф., заведующий кафедрой «Системы автоматизированного проектирования вычислительных средств»;

Костров Б.В., д.т.н., проф., заведующий кафедрой «Электронные вычислительные машины»;

Кошелев В.И., д.т.н., проф., заведующий кафедрой «Радиотехнические системы»;

Лукьянова Г.С., к. ф.-м.н., доцент кафедры «Высшая математика»;

Паршин Ю.Н., д.т.н., проф., заведующий кафедрой «Радиотехнические устройства»;

Круглов С.А., к.т.н., доц., заведующий кафедрой «Промышленная электроника»;

Мусолин А.К., д.т.н., проф., заведующий кафедрой «Автоматизация информационных и технологических процессов»;

Перфильев С.В., д.э.н., проф., заведующий кафедрой «Государственное, муниципальное и корпоративное управление»;

Овечкин Г.В., д.т.н., профессор кафедры «Вычислительная и прикладная математика»;

Пржегорлинский В.Н., к.т.н., доц., заведующий кафедрой «Информационная безопасность»;

Пылькин А.Н., д.т.н., проф., декан факультета Вычислительной техники, заведующий кафедрой «Вычислительная и прикладная математика»;

Серебряков А.Е., к.т.н., зам. заведующего кафедрой «Электронные приборы»;

Соколов А.С., д.и.н., заведующий кафедрой «Истории, философии и права»;

Таганов А.И., д.т.н., проф., заведующий кафедрой «Космические технологии»;

Федорова Л.А., д.э.н., профессор кафедры «Государственное, муниципальное и корпоративное управление»;

Холомина Т.А., д.ф.-м.н., проф., заведующая кафедрой «Микро- и наноэлектроника»;

Чеглакова С.Г., д.э.н, проф., заведующая кафедрой «Экономическая безопасность, анализ и учет».

**МЕЖДУНАРОДНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ
«СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ.
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ»**

**СЕКЦИЯ «АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ»**

УДК 519.816; ГРНТИ 27.47.19

ВАЛЕНТНОСТЬ ФУНКЦИЙ ОДИНОЧНО-ПУСТОГО ВЫБОРА

И.А. Цветков

*Рязанский государственный радиотехнический университет,
Россия, Рязань, 19197113@rambler.ru*

Аннотация. Предлагаются понятия: соседняя функция одиночно-пустого выбора, валентность функции одиночно-пустого выбора. Приводятся доказанные автором утверждения о валентности функций одиночно-пустого выбора.

Ключевые слова. Валентность, функция выбора, функция одиночно-пустого выбора.

VALENCE OF SINGLE-EMPTY CHOICE FUNCTIONS

I.A. Tsvetkov

*Ryazan State Radio Engineering University,
Russia, Ryazan, 19197113@rambler.ru*

Abstract. The concepts: the neighboring function of a single-empty choice function, the valence of a single-empty choice function are proposed. The statements which proved by the author about the valence of single-empty choice functions are given.

Keywords. Valence, choice function, single-empty choice function.

Введение

Функция одиночно-пустого выбора (это понятие предложено в [2, 3]) ставит в соответствие каждому непустому подмножеству заданного множества с двумя или более элементами (множества вариантов) одно подмножество этого подмножества: или *1-элементное* (*одиночный* выбор), или *пустое* (*пустой* выбор). Причем выбор *одиночный не для всех* непустых подмножеств множества вариантов, выбор *пустой не для всех* таких подмножеств.

Функция одиночно-пустого выбора — частный случай *функции выбора* [1], которая ставит в соответствие каждому непустому подмножеству множества вариантов *одно* из подмножеств этого подмножества (возможно, пустое).

В докладе для функций одиночно-пустого выбора вводятся два новых понятия.

Первое понятие — *соседняя функция одиночно-пустого выбора*: функция одиночно-пустого выбора — соседняя для другой функции одиночно-пустого выбора, заданной на том же множестве вариантов, если и только если значения этих двух функций *различны ровно для одного* непустого подмножества этого множества.

Второе понятие — *валентность функции одиночно-пустого выбора*: число всех различных *соседних* для нее функций одиночно-пустого выбора.

В докладе приводятся доказанные его автором утверждения о валентности функций одиночно-пустого выбора, заданных на произвольных (имеющих мощность более единицы) множествах вариантов.

Обозначения и термины

Обозначим: $|S|$ — мощность множества S ; $A_n = \{a_1, a_2, \dots, a_n\}$ — множество вариантов, где $n \in N_2$; $N_2 = N \setminus \{1\}$; $N_3 = N \setminus \{1, 2\}$; $N = \{1, 2, \dots\}$ — множество натуральных чисел; B_n — семейство всех непустых подмножеств множества A_n .

Функция одиночно-пустого выбора C_{A_n} , $n \in N_2$, заданная на множестве вариантов A_n , определяется так: $C_{A_n}(X) \subseteq X$ и $|C_{A_n}(X)| \in \{0, 1\}$ при всех $X \in B_n$, причем существуют $Y \in B_n$ и $Z \in B_n$, для которых $|C_{A_n}(Y)| = 1$ и $C_{A_n}(Z) = \emptyset$.

Обозначим при $n \in N_2$: E_n — множество всех функций одиночно-пустого выбора, заданных на множестве A_n ; $I_n = \{1, 2, \dots, |E_n|\}$. Имеем (см. [2]) для всех $n \in N_2$:

$$|E_n| = \prod_{k=1}^n (k+1) C_n^k - \prod_{m=1}^n m C_n^m - 1. \quad (1)$$

Пусть $C_{A_n}^{(1)} \in E_n$ и $C_{A_n}^{(2)} \in E_n$ — различные функции одиночно-пустого выбора, заданные на множестве вариантов A_n , $n \in N_2$. Согласно формуле (1) при каждом $n \in N_2$ имеем $|E_n| > 1$. Следовательно, для всех $n \in N_2$ указанные функции $C_{A_n}^{(1)}$ и $C_{A_n}^{(2)}$ существуют.

Определение 1. Функция $C_{A_n}^{(2)} \in E_n$ — **соседняя** для функции $C_{A_n}^{(1)} \in E_n$, если и только если $C_{A_n}^{(2)}(Y) \neq C_{A_n}^{(1)}(Y)$ при единственном $Y \in B_n$ и $C_{A_n}^{(2)}(X) = C_{A_n}^{(1)}(X)$ при каждом $X \in B_n \setminus \{Y\}$, где $n \in N_2$.

Если функция $C_{A_n}^{(2)} \in E_n$ соседняя для функции $C_{A_n}^{(1)} \in E_n$, то применяем обозначение $C_{A_n}^{(2)} \parallel C_{A_n}^{(1)}$, иначе — обозначение $C_{A_n}^{(2)} \perp C_{A_n}^{(1)}$, где $n \in N_2$. Из определения 1 следует: при $C_{A_n}^{(2)} \parallel C_{A_n}^{(1)}$, $n \in N_2$, значения функций $C_{A_n}^{(2)} \in E_n$ и $C_{A_n}^{(1)} \in E_n$ различаются ровно для одного значения аргумента из B_n . Таким образом, имеет место $C_{A_n}^{(2)} \parallel C_{A_n}^{(1)}$, если и только если $\left| \left\{ X : X \in B_n, C_{A_n}^{(2)}(X) = C_{A_n}^{(1)}(X) \right\} \right| = |B_n| - 1$, где $n \in N_2$. Из определения 1 следует: для каждого $n \in N_2$ при $C_{A_n}^{(2)} \parallel C_{A_n}^{(1)}$ имеем $C_{A_n}^{(1)} \parallel C_{A_n}^{(2)}$, при $C_{A_n}^{(2)} \perp C_{A_n}^{(1)}$ имеем $C_{A_n}^{(1)} \perp C_{A_n}^{(2)}$.

Пример. В табл. 1 приведены значения всех девяти — по формуле (1) получим $|E_2| = 9$ — функций одиночно-пустого выбора $C_{A_2}^{(i)} \in E_2$, $i \in I_2$, $I_2 = \{1, 2, \dots, 9\}$, заданных на множестве вариантов $A_2 = \{a_1, a_2\}$. Для значений аргументов X , указанных в табл. 1, имеем $X \in B_2$, где $B_2 = \{\{a_1\}, \{a_2\}, \{a_1, a_2\}\}$.

Таблица 1. Функции одиночно-пустого выбора, заданные на множестве A_2

X	$C_{A_2}^{(1)}(X)$	$C_{A_2}^{(2)}(X)$	$C_{A_2}^{(3)}(X)$	$C_{A_2}^{(4)}(X)$	$C_{A_2}^{(5)}(X)$	$C_{A_2}^{(6)}(X)$	$C_{A_2}^{(7)}(X)$	$C_{A_2}^{(8)}(X)$	$C_{A_2}^{(9)}(X)$
$\{a_1\}$	$\{a_1\}$	$\emptyset$	$\{a_1\}$	$\emptyset$	$\{a_1\}$	$\emptyset$	$\emptyset$	$\{a_1\}$	$\emptyset$
$\{a_2\}$	$\emptyset$	$\{a_2\}$	$\{a_2\}$	$\emptyset$	$\emptyset$	$\{a_2\}$	$\emptyset$	$\emptyset$	$\{a_2\}$
$\{a_1, a_2\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{a_1\}$	$\{a_1\}$	$\{a_1\}$	$\{a_2\}$	$\{a_2\}$	$\{a_2\}$

Имеем:

$$C_{A_2}^{(3)}(\{a_1\}) \neq C_{A_2}^{(2)}(\{a_1\}), C_{A_2}^{(3)}(\{a_2\}) = C_{A_2}^{(2)}(\{a_2\}), C_{A_2}^{(3)}(\{a_1, a_2\}) = C_{A_2}^{(2)}(\{a_1, a_2\}).$$

Поэтому функция $C_{A_2}^{(3)}$ — *соседняя* для функции $C_{A_2}^{(2)}$, т.е. $C_{A_2}^{(3)} \parallel C_{A_2}^{(2)}$; следовательно, $C_{A_2}^{(2)} \parallel C_{A_2}^{(3)}$.

$$\text{Обозначим } R_{32} = \left\{ X : X \in B_2, C_{A_2}^{(3)}(X) = C_{A_2}^{(2)}(X) \right\} = \{\{a_2\}, \{a_1, a_2\}\}.$$

Тогда $|R_{32}| = 2$. Равенство $|R_{32}| = |B_2| - 1$ истинно, поскольку $|B_2| = 3$.

Имеем:

$$C_{A_2}^{(3)}(\{a_1\}) = C_{A_2}^{(5)}(\{a_1\}), C_{A_2}^{(3)}(\{a_2\}) \neq C_{A_2}^{(5)}(\{a_2\}), C_{A_2}^{(3)}(\{a_1, a_2\}) \neq C_{A_2}^{(5)}(\{a_1, a_2\}).$$

Поэтому функция $C_{A_2}^{(3)}$ — *не соседняя* для функции $C_{A_2}^{(5)}$, т.е. $C_{A_2}^{(3)} \perp C_{A_2}^{(5)}$; следовательно, $C_{A_2}^{(5)} \perp C_{A_2}^{(3)}$.

$$\text{Обозначим } R_{35} = \left\{ X : X \in B_2, C_{A_2}^{(3)}(X) = C_{A_2}^{(5)}(X) \right\} = \{\{a_1\}\}.$$

Тогда $|R_{35}| = 1$. Равенство $|R_{35}| = |B_2| - 1$ ложно.

Определение 2. Валентность функции одиночно-пустого выбора — это число *всех соседних* для нее функций одиночно-пустого выбора.

Определение 3. k -валентная функция одиночно-пустого выбора — это функция, валентность которой равна k .

Обозначим $V\left(C_{A_n}\right)$ валентность функции $C_{A_n} \in E_n$, где $n \in N_2$. Согласно определению 2 имеем $V\left(C_{A_n}\right) = \left| \left\{ C_{A_n}^{(*)} : C_{A_n}^{(*)} \in E_n, C_{A_n}^{(*)} \parallel C_{A_n} \right\} \right|$, $n \in N_2$.

Обозначим для $n \in N_2$: $g(n) = n \cdot 2^{n-1}$, $W_n = \{g(n)-n, g(n)-n+1, \dots, g(n)-1\}$. Определим функцию $J: J(a, b) = 1$ при $a = b$, $J(a, b) = 0$ при $a \neq b$, где $a \in N$, $b \in N$. Обозначим $T_n(v) = \left| \left\{ C_{A_n} : C_{A_n} \in E_n, V\left(C_{A_n}\right) = v \right\} \right|$, $n \in N_2$, число всех функций одиночно-пустого выбора, заданных на множестве A_n и имеющих валентность v .

Пример. По табл. 1 найдем для каждого $i \in I_2$ множество Q_i всех функций одиночно-пустого выбора, соседних для функции $C_{A_2}^{(i)} \in E_2$, где $I_2 = \{1, 2, \dots, 9\}$. Учитываем, что $|E_2| = 9$ согласно формуле (1). Имеем $Q_i = \left\{ C_{A_2}^{(j)} : C_{A_2}^{(j)} \in E_2, j \in I_2 \setminus \{i\}, C_{A_2}^{(j)} \parallel C_{A_2}^{(i)} \right\}$, $i \in I_2$.

$$\text{Получим: } Q_1 = \left\{ C_{A_2}^{(3)}, C_{A_2}^{(5)}, C_{A_2}^{(8)} \right\}, Q_2 = \left\{ C_{A_2}^{(3)}, C_{A_2}^{(6)}, C_{A_2}^{(9)} \right\}, Q_3 = \left\{ C_{A_2}^{(1)}, C_{A_2}^{(2)} \right\},$$

$$Q_4 = \left\{ C_{A_2}^{(5)}, C_{A_2}^{(6)}, C_{A_2}^{(7)} \right\}, Q_5 = \left\{ C_{A_2}^{(1)}, C_{A_2}^{(4)}, C_{A_2}^{(8)} \right\}, Q_6 = \left\{ C_{A_2}^{(2)}, C_{A_2}^{(4)}, C_{A_2}^{(9)} \right\},$$

$$Q_7 = \left\{ C_{A_2}^{(4)}, C_{A_2}^{(8)}, C_{A_2}^{(9)} \right\}, Q_8 = \left\{ C_{A_2}^{(1)}, C_{A_2}^{(5)}, C_{A_2}^{(7)} \right\}, Q_9 = \left\{ C_{A_2}^{(2)}, C_{A_2}^{(6)}, C_{A_2}^{(7)} \right\}.$$

Тогда $V\left(C_{A_2}^{(i)}\right) = |Q_i|$, $i \in I_2$. Имеем: $V\left(C_{A_2}^{(j)}\right) = 3$ при $j \in I_2 \setminus \{3\}$; $V\left(C_{A_2}^{(3)}\right) = 2$.

Основные результаты

Автором доклада доказаны следующие утверждения о валентности функций одиночно-пустого выбора.

Утверждение 1. Для каждой функции одиночно-пустого выбора $C_{A_2} \in E_2$ имеет место $V\left(C_{A_2}\right) \in W_2$.

Утверждение 2. Имеет место: $T_2(2) = 1$, $T_2(3) = 8$.

Утверждение 3. При любом $n \in N_3$ для каждой функции одиночно-пустого выбора $C_{A_n} \in E_n$ имеет место $V\left(C_{A_n}\right) \in W_n \cup \{g(n)\}$.

Утверждение 4. При любом $n \in N_3$ имеет место:

$$T_n(v) = J(v, g(n)-1)g(n) + C_n^{g(n)-v} \prod_{k=1}^n C_n^k - J(k, g(n)-v)$$

для $v \in W_n$; $T_n(g(n)) = |E_n| - \sum_{v=g(n)-n}^{g(n)-1} T_n(v)$, где $|E_n|$ определяется формулой (1).

Пример. Для множества вариантов $A_2 = \{a_1, a_2\}$ имеем: $g(2) = 4$, $W_2 = \{2, 3\}$, $|E_2| = 9$. Определим матрицу M_2 размером $|E_2| \times |E_2| = 9 \times 9$ с такими элементами $m_{ij} \in \{0, 1\}$, $i = 1, 2, \dots, 9$, $j = 1, 2, \dots, 9$, что $m_{ij} = 1$ при $C_{A_2}^{(j)} \parallel C_{A_2}^{(i)}$, $m_{ij} = 0$ при $C_{A_2}^{(j)} \perp C_{A_2}^{(i)}$, где $C_{A_2}^{(k)} \in E_2$, $k = 1, 2, \dots, 9$, — функция одиночно-пустого выбора, значения которой заданы в табл. 1. Учитываем, что $C_{A_2}^{(t)} \perp C_{A_2}^{(t)}$ при каждом $t = 1, 2, \dots, 9$ в силу определения 1. Имеем

$$M_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

Число единиц в каждой строке матрицы M_2 (и в каждом ее столбце, поскольку M_2 — симметрическая матрица) принадлежит множеству W_2 . Ровно в одной строке матрицы M_2 число единиц равно двум. Имеем $T_2(2) = 1$. В каждой из остальных восьми строк матрицы M_2 число единиц равно трем. Имеем $T_2(3) = 8$.

Таким образом, в множестве E_2 есть ровно одна 2-валентная функция одиночно-пустого выбора и ровно восемь 3-валентных функций одиночно-пустого выбора.

Пример. Для множества вариантов $A_3 = \{a_1, a_2, a_3\}$ имеем: $g(3) = 12$, $W_3 = \{9, 10, 11\}$, $|E_3| = 839$. В силу утверждения 3 валентность каждой функции из E_3 принадлежит множеству $W_3 \cup \{g(3)\} = \{9, 10, 11, 12\}$. В табл. 2 приведены вычисленные в соответствии с утверждением 4 значения числа функций из множества E_3 , имеющих заданную валентность.

Таблица 2. Число функций из множества E_3 с заданной валентностью

Валентность	9	10	11	12
Число функций	8	36	84	711

Библиографический список

1. Айзерман М.А., Алескерев Ф.Т. Выбор вариантов: основы теории. — М.: Наука, 1990. — 240 с.
2. Цветков И.А. Условия Чернова для функций одиночно-пустого выбора // Вестник РГРТУ. — 2012. — № 1 (39). — Ч. 2. — С. 57–64.
3. Цветков И.А. Функции одиночно-пустого выбора // Перспективы развития гуманитарных и технических систем: материалы всероссийской науч. конф. — Ч. 2. — Таганрог: изд-во ТТИ ЮФУ, 2011. — С. 77–79.

УДК 519.816; ГРНТИ 27.47.19

ФУНКЦИИ РАЗБИВНОГО ВЫБОРА

И.А. Цветков

*Рязанский государственный радиотехнический университет,
Россия, Рязань, 19197113@rambler.ru*

Аннотация. Предлагается понятие функции разбивного выбора (частного случая функции одиночно-пустого выбора). Для функций разбивного выбора исследуются условия Чернова. Определяется число таких функций.

Ключевые слова. Разбиение множества, условие Чернова, функция выбора.

PARTITIONED CHOICE FUNCTIONS

I.A. Tsvetkov

*Ryazan State Radio Engineering University,
Russia, Ryazan, 19197113@rambler.ru*

Abstract. The concept of a partitioned choice function (a special case of a single-empty choice function) is proposed. For partitioned choice functions the Chernoff's conditions are investigated and the number of such functions is determined.

Keywords. Partition of a set, Chernoff's condition, choice function.

Введение

Функция выбора C_{A_n} на множестве вариантов A_n , $n \in N_2$, ставит в соответствие каждому значению аргумента $X \in B_n$ значение функции $C_{A_n}(X) \subseteq X$ [1]. Здесь и далее $A_n = \{a_1, a_2, \dots, a_n\}$; $N_2 = N \setminus \{1\}$, где $N = \{1, 2, \dots\}$ — множество натуральных чисел; B_n — семейство всех непустых подмножеств множества A_n .

В [2, 3] введено понятие функции одиночно-пустого выбора C_{A_n} , $n \in N_2$, для которой $|C_{A_n}(X)| \in \{0, 1\}$ при каждом $X \in B_n$ и существуют такие $Y \in B_n$ и $Z \in B_n$, что $|C_{A_n}(Y)| = 1$, $C_{A_n}(Z) = \emptyset$. Здесь и далее $|S|$ — мощность множества S .

В докладе предлагается понятие функции разбивного выбора — частного случая функции одиночно-пустого выбора. Для функций разбивного выбора исследуются условия Чернова. Определяется число таких функций.

Обозначения и термины

Обозначим D_n , $n \in N_2$, семейство всех неупорядоченных разбиений множества A_n на непустые подмножества. Для каждого $P \in D_n$ и $n \in N_2$ функция разбиного выбора $C_{A_n}^{(P)}$ определена так: $|C_{A_n}^{(P)}(Y)| = 1$ при всех $Y \in P$, $C_{A_n}^{(P)}(Z) = \emptyset$ при всех $Z \in B_n \setminus P$.

Пример. Для множества вариантов $A_3 = \{a_1, a_2, a_3\}$ рассмотрим разбиение $P = \{\{a_2\}, \{a_1, a_3\}\}$ из D_3 . Имеем $B_3 \setminus P = \{\{a_1\}, \{a_3\}, \{a_1, a_2\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}$. Для этого P есть ровно две функции разбиного выбора. Значения одной из них: $C_{A_3}^{(P)}(\{a_2\}) = \{a_2\}$, $C_{A_3}^{(P)}(\{a_1, a_3\}) = \{a_1\}$, $C_{A_3}^{(P)}(Z) = \emptyset$ при всех $Z \in B_3 \setminus P$. Другая имеет те же значения, кроме $\{a_3\}$ для значения аргумента $\{a_1, a_3\}$.

«Качество» функции выбора характеризуется, в частности, выполнением условий Чернова — условия наследования, условия константности, условия согласия, условия независимости от отбрасывания отвергнутых вариантов (кратко — условия независимости) [1].

Для функции выбора C_{A_n} на множестве вариантов A_n , $n \in N_2$, выполняются условия Чернова [1]: 1) условие наследования, если и только если для любого $X \in B_n$ и любого непустого $Y \subset X$ имеет место

$$Y \cap C_{A_n}(X) \subseteq C_{A_n}(Y); \quad (1)$$

2) условие константности, если и только если для любого $X \in B_n$ и любого непустого $Y \subset X$ из $C_{A_n}(X) = \emptyset$ следует $C_{A_n}(Y) = \emptyset$, а из $Y \cap C_{A_n}(X) \neq \emptyset$ следует $Y \cap C_{A_n}(X) = C_{A_n}(Y)$; 3) условие согласия, если и только если для любых $X \in B_n$ и $Y \in B_n$ имеет место

$$C_{A_n}(X) \cap C_{A_n}(Y) \subseteq C_{A_n}(X \cup Y); \quad (2)$$

4) условие независимости, если и только если для любого $X \in B_n$ и любого непустого $Y \subset X$ при $C_{A_n}(X) \subseteq Y$ имеет место $C_{A_n}(X) = C_{A_n}(Y)$.

Называем *конфигурацией условий* строку из четырех символов, каждый из которых равен 0 или 1 и соответствует невыполнению (символ 0) и выполнению (символ 1) условий Чернова в порядке слева направо: наследования, константности, согласия, независимости.

Обозначим $B_n^{(1)}$, $n \in N_2$, семейство всех 1-элементных подмножеств множества A_n . При любом $n \in N_2$ имеем $B_n^{(1)} \in D_n$. Обозначим $B_n^{(*)} = B_n \setminus B_n^{(1)}$, $n \in N_2$, семейство всех подмножеств множества A_n , имеющих мощность больше единицы.

Символ ♦ показывает окончание доказательства.

Основные результаты

Автором доклада доказаны следующие утверждения о функциях разбивного выбора.

Утверждение 1. При любом $n \in N_2$ есть ровно одна функция разбивного выбора $C_{A_n}^{(P)}$, для которой выполняется условие наследования, где $P = B_n^{(1)}$.

Утверждение 2. При любом $n \in N_2$ и каждом $P \in D_n$ для всех функций разбивного выбора $C_{A_n}^{(P)}$, определенных при этом P , выполняется условие согласия.

Утверждение 3. При любом $n \in N_2$ и каждом $P \in D_n$ для всех функций разбивного выбора $C_{A_n}^{(P)}$, определенных при этом P , не выполняется условие независимости.

Утверждение 4. При любом $n \in N_2$ и каждом $P \in D_n$ для всех функций разбивного выбора $C_{A_n}^{(P)}$, определенных при этом P , не выполняется условие константности.

Утверждение 5. При любом $n \in N_2$ и каждом таком $P \in D_n$, что $P = \{X_1, X_2, \dots, X_k\}$, $X_i \in B_n$, $i = 1, 2, \dots, k$, $1 \leq k \leq n$, число всех функций разбивного выбора $C_{A_n}^{(P)}$, определенных при этом P , равно $\prod_{i=1}^k |X_i|$.

Утверждение 6. При любом $n \in N_2$ число всех функций разбивного выбора на множестве A_n равно $F_n = 1 + \sum_{i=1}^{n-1} (n-i)^i C_n^i$.

Из утверждений 1 – 4 следует, что при любом $n \in N_2$: 1) каждая функция разбивного выбора на множестве A_n имеет одну из двух конфигураций условий — 0010 или 1010; 2) есть ровно одна функция разбивного выбора на множестве A_n с конфигурацией условий 1010, остальные $F_n - 1$ функций разбивного выбора имеют конфигурацию условий 0010.

Пример. Для множества вариантов $A_3 = \{a_1, a_2, a_3\}$ рассмотрим разбиение $P_1 = \{\{a_1\}, \{a_2\}, \{a_3\}\}$ из D_3 . Тогда $C_{A_3}^{(P_1)}(\{a_i\}) = \{a_i\}$ для $i = 1, 2, 3$; $C_{A_3}^{(P_1)}(Z) = \emptyset$ при всех $Z \in B_3 \setminus P_1$. Здесь $B_3 \setminus P_1 = \{\{a_1, a_2\}, \{a_1, a_3\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}$. Так как $P_1 = B_3^{(1)}$, в силу утверждений 1 – 4 функция разбивного выбора $C_{A_3}^{(P_1)}$ имеет конфигурацию условий 1010.

Пример. Для множества вариантов $A_3 = \{a_1, a_2, a_3\}$ рассмотрим разбиение $P_2 = \{\{a_2\}, \{a_1, a_3\}\}$ из D_3 . Тогда $C_{A_3}^{(P_2)}(\{a_2\}) = \{a_2\}$, $C_{A_3}^{(P_2)}(Z) = \emptyset$ при всех $Z \in B_3 \setminus P_2$. Здесь $B_3 \setminus P_2 = \{\{a_1\}, \{a_3\}, \{a_1, a_2\}, \{a_2, a_3\}, \{a_1, a_2, a_3\}\}$. Пусть

$C_{A_3}^{(P_2)}(\{a_1, a_3\}) = \{a_3\}$. Так как $P_2 \neq B_3^{(1)}$ (где $B_3^{(1)} = \{\{a_1\}, \{a_2\}, \{a_3\}\}$), в силу утверждений 1 – 4 функция разбивного выбора $C_{A_3}^{(P_2)}$ имеет конфигурацию условий 0010.

Пример. Значения F_n числа всех функций разбивного выбора на множестве A_n , полученные для некоторых значений n на основании утверждения 6, указаны в табл 1.

Таблица 1. Число всех функций разбивного выбора на множестве A_n

n	2	3	4	5	6	7	8	9	10
F_n	3	10	41	196	1 057	6 322	41 393	293 608	2 237 921

Доказательства

Для примера приведем полученные автором доклада доказательства утверждений 1 и 2.

Доказательство утверждения 1. Для произвольного $n \in N_2$ рассмотрим 2 случая.

Случай 1. Пусть $P = B_n^{(1)}$. Тогда $P \in D_n$. Для таких X и Y , что $X \in B_n$, $Y \in B_n$, $Y \subset X$, рассмотрим 2 случая.

Случай 1А. Пусть $Y \in P$. Имеем $|Y| = 1$. Поэтому из $Y \subset X$ следует $|X| > 1$. Тогда $X \in B_n^{(*)}$ и

$$C_{A_n}^{(P)}(X) = \emptyset \quad (3)$$

(поскольку $X \notin P$). При этом

$$Y \cap C_{A_n}^{(P)}(X) = \emptyset. \quad (4)$$

Следовательно, для функции $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (1).

Случай 1Б. Пусть $Y \notin P$. Тогда $Y \in B_n^{(*)}$. Имеем $|Y| > 1$. Если $Y \subset X$, то $|X| > 1$. Следовательно, $X \in B_n^{(*)}$. Тогда истинны: равенство (3) (поскольку $X \notin P$) и равенство (4). Следовательно, для функции $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (1).

Анализ случаев 1А и 1Б показывает, что при любом $n \in N_2$ для функции разбивного выбора $C_{A_n}^{(P)}$, где $P = B_n^{(1)}$, условие наследования выполняется.

Случай 2. Пусть $P \neq B_n^{(1)}$, $P \in D_n$. Тогда $P \in D_n \setminus \{B_n^{(1)}\}$. Следовательно, существует $X \in P$, для которого $|X| > 1$ и $C_{A_n}^{(P)}(X) = \{a\}$, где $a \in A_n$. Тогда $a \in X$, так как

$C_{A_n}^{(P)}(X) \subseteq X$. Пусть $Y = \{a\}$. При этом $Y \subset X$ (учитываем, что $|Y| = 1$, $|X| > 1$), $X \cap Y \neq \emptyset$ и из $X \in P$ следует: $Y \notin P$, $C_{A_n}^{(P)}(Y) = \emptyset$. Получим $Y \cap C_{A_n}^{(P)}(X) = \{a\} \cap \{a\} = \{a\}$. Следовательно, для функции $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) не имеет места (1). Поэтому при любом $n \in N_2$ и каждом таком $P \in D_n$, что $P \neq B_n^{(1)}$, для функции разбивного выбора $C_{A_n}^{(P)}$, определенной при этом P , условие наследования не выполняется.

Анализ случаев 1 и 2 показывает, что при любом $n \in N_2$ *есть* функция разбивного выбора $C_{A_n}^{(P)}$ (где $P = B_n^{(1)}$), для которой условие наследования *выполняется*. При каждом $n \in N_2$ это разбиение P *единственное*. Ему принадлежат только такие подмножества множества A_n , которые имеют *единичную* мощность. Для каждого из них имеется *ровно один* способ выбрать значение функции $C_{A_n}^{(P)}$ (оно равно этому подмножеству). Следовательно, при любом $n \in N_2$ для $P = B_n^{(1)}$ *есть ровно одна* функция разбивного выбора $C_{A_n}^{(P)}$, определенная при этом P , для которой условие наследования *выполняется*. При любом $n \in N_2$ и каждом таком $R \in D_n$, что $R \neq B_n^{(1)}$, для *всех* функций разбивного выбора $C_{A_n}^{(R)}$, определенных при этом R , условие наследования *не выполняется*.

Таким образом, при любом $n \in N_2$ *есть ровно одна* функция разбивного выбора $C_{A_n}^{(P)}$, для которой условие наследования *выполняется*. Здесь $P = B_n^{(1)}$. ♦

Доказательство утверждения 2. Для произвольных $n \in N_2$ и $P \in D_n$ рассмотрим 4 случая.

Случай 1. Пусть $X \in P$, $Y \in P$. Тогда $X \cap Y = \emptyset$. Поэтому

$$C_{A_n}^{(P)}(X) \cap C_{A_n}^{(P)}(Y) = \emptyset, \quad (5)$$

так как $C_{A_n}^{(P)}(X) \subseteq X$, $C_{A_n}^{(P)}(Y) \subseteq Y$. Следовательно, для функции разбивного выбора $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (2).

Случай 2. Пусть $X \in P$, $Y \in B_n \setminus P$. Тогда $C_{A_n}^{(P)}(Y) = \emptyset$ и имеет место (5). Следовательно, для функции разбивного выбора $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (2).

Случай 3. Пусть $X \in B_n \setminus P$, $Y \in P$. Тогда $C_{A_n}^{(P)}(X) = \emptyset$ и имеет место (5). Следовательно, для функции разбивного выбора $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (2).

Случай 4. Пусть $X \in B_n \setminus P$, $Y \in B_n \setminus P$. Тогда $C_{A_n}^{(P)}(X) = \emptyset$ и имеет место (5). Следовательно, для функции разбивного выбора $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (2).

Анализ случаев 1 – 4 показывает, что при любом $n \in N_2$, каждом $P \in D_n$ и всех $X \in B_n$, $Y \in B_n$ для функции разбивного выбора $C_{A_n}^{(P)}$ (при замене C_{A_n} на $C_{A_n}^{(P)}$) имеет место (2). Следовательно, при любом $n \in N_2$ и каждом $P \in D_n$ для всех функций разбивного выбора $C_{A_n}^{(P)}$, определенных при этом P , условие согласия выполняется. ♦

Библиографический список

1. Айзерман М.А., Алескерев Ф.Т. Выбор вариантов: основы теории. — М.: Наука, 1990. — 240 с.
2. Цветков И.А. Условия Чернова для функций одиночно-пустого выбора // Вестник РГРТУ. — 2012. — № 1 (39). — Ч. 2. — С. 57–64.
3. Цветков И.А. Функции одиночно-пустого выбора // Перспективы развития гуманитарных и технических систем: материалы всероссийской науч. конф. — Ч. 2. — Таганрог: изд-во ТТИ ЮФУ, 2011. — С. 77–79.

УДК 378.147:004; ГРНТИ 82.33.13

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ЦЕНТРА ПОДГОТОВКИ ВОЛОНТЕРОВ

В.В. Коваленко*, Р.К. Минасян**

* Сочинский Государственный Университет,

** Московский технологический университет МИРЭА

Россия, Сочи, vlvas@mail.ru

Аннотация. Рассматривается разработка информационной системы для центра подготовки волонтеров на базе принципов процессного подхода к управлению. Основу системы составляют бизнес-процессы, обеспечивающие достижение заданных в стратегической карте подцелей и значений ключевых показателей.

Ключевые слова. Информационная система, процессный подход, бизнес-процесс, Business Studio, волонтер

DEVELOPMENT OF INFORMATION SYSTEM TRAINING CENTER FOR VOLUNTEERS

V. V. Kovalenko*, R. K. Minasyan**

* Sochi State University,

** Moscow Technological University (MIREA)

Russia, Sochi, vlvas@mail.ru

Abstract. Development of information system for the training center volunteers on the basis of the principles of process approach to management is considered. The basis of system is realized by the business processes providing achievement of the local goals set in a strategic map and values of key indicators.

Keywords. Information system, process approach, Business-process, Business Studio, volunteer

В настоящее время волонтерская деятельность развивается бурными темпами и набирает большую популярность среди студентов образовательных учреждений. Сейчас невозможно представить, какое-либо городское культурно-массовое, социально значимое, экологическое, гражданско-патриотическое мероприятие без участия волонтеров.

Во время проведения XXII Олимпийских зимних игр и XI Паралимпийских зимних игр в Сочи 2014 года, Центром подготовки волонтеров (ЦПВ) «Forward» Сочинского государственного университета было подготовлено 1339 студентов и 54 преподавателя, которые отработали более 25000 смен. Кроме этого, волонтеры СГУ работают на «домашних» играх Хоккейного клуба «Сочи». Работали на крупных международных мероприятиях как Гран-При России по кольцевым автогонкам Formula 1 и Хоровые Игры 2016. В будущем волонтеры будут работать на Кубке Конфедераций по футболу в 2017 году и Чемпионате мира по футболу в 2018 году.

Из-за большого количества мероприятий сотрудники ЦПВ встречаются с рядом проблем по набору волонтеров, контролю их обучения и управления за выходом на смены. Поэтому без информационной системы, в которой будет храниться архив волонтерской деятельности по каждому студенту и преподавателю, не обойтись, что определяет актуальность работы.

Сейчас ЦПВ вышел практически на профессиональный уровень и работает по утвержденной ректоратом схеме. Все заявки от организаций рассматривает ректор, который принимает решение об их исполнении. Затем общий отдел оформляет необходимые документы для передачи в волонтерский центр, где сотрудники ЦПВ приступают к работе с заказчиком. При этом возникает проблема хранения, поиска, и формирования необходимой документации во многом из-за предоставления части документов в бумажном виде.

Разработанная информационная система позволяет избавиться от указанных выше проблем за счёт учета, хранения и обработки информации о волонтерской деятельности каждого студента и преподавателя, а также в результате частичной автоматизации информационно-аналитической деятельности сотрудников волонтерского центра.

Оригинальность работы определяется тем, что функциональная модель разрабатывалась на основе стратегической карты, в которой стратегическая цель определена как повышение социальной активности среди студентов и преподавательского состава университета. Для стратегической карты была реализована декомпозиция стратегической цели на подцели, для которых были заданы показатели, определяющие степень достижения целей (табл. 1).

Таблица 1. Показатели из стратегической карты

№ п/п	Наименование цели	Показатель
1	Повысить социальную активность среди студентов	% волонтеров среди студентов
2	Повысить % волонтеров	% привлеченных волонтеров
3	Увеличить % волонтеров, вышедших на смены	% волонтеров, которые вышли на смены
4	Увеличить % отработанных часов	Часы, отработанные на мероприятии
5	Качественно обучать волонтеров	% волонтеров, успешно прошедших обучение
6	Увеличить % волонтеров, успешно сдавших сессию	% волонтеров, успешно сдавших сессию
7	Уменьшить % отказов от мероприятия	% волонтеров, отказавшихся о мероприятия

Все показатели закреплены за процессами, которые обеспечивают достижение их заданных значений. При этом значения показателей для всех временных интервалов хранятся в

базе данных, что позволяет не только выполнить их мониторинг, но и реализовать на их основе управление бизнес-процессами в соответствии с циклом Деминга.

Всего в проекте реализовано несколько бизнес-процессов, таких как: «Набор волонтеров», «Работа с волонтерами (обучение)», «Работа с волонтерами (утвержденные позиции)», «Работа с волонтерами (смены)», «Анализ волонтерской деятельности». В соответствии со стратегической картой модернизирован бизнес-процесс «Набор волонтеров» и разработан новый бизнес-процесс «Работа с волонтерами (утвержденные позиции)», который ранее не использовался в работе ЦПВ.

Бизнес-процесс «Набор волонтеров» обеспечивает выполнение показателя «Повысить % волонтеров» и является одним из базовых в работе ЦПВ (рис. 1). Дело в том, что после Олимпийских игр возрос спрос на волонтеров во многих мероприятиях и сферах деятельности администрации города Сочи. Поэтому формирование группы волонтеров для каждого нового мероприятия является не только ответственной работой, но и достаточно трудоемкой.

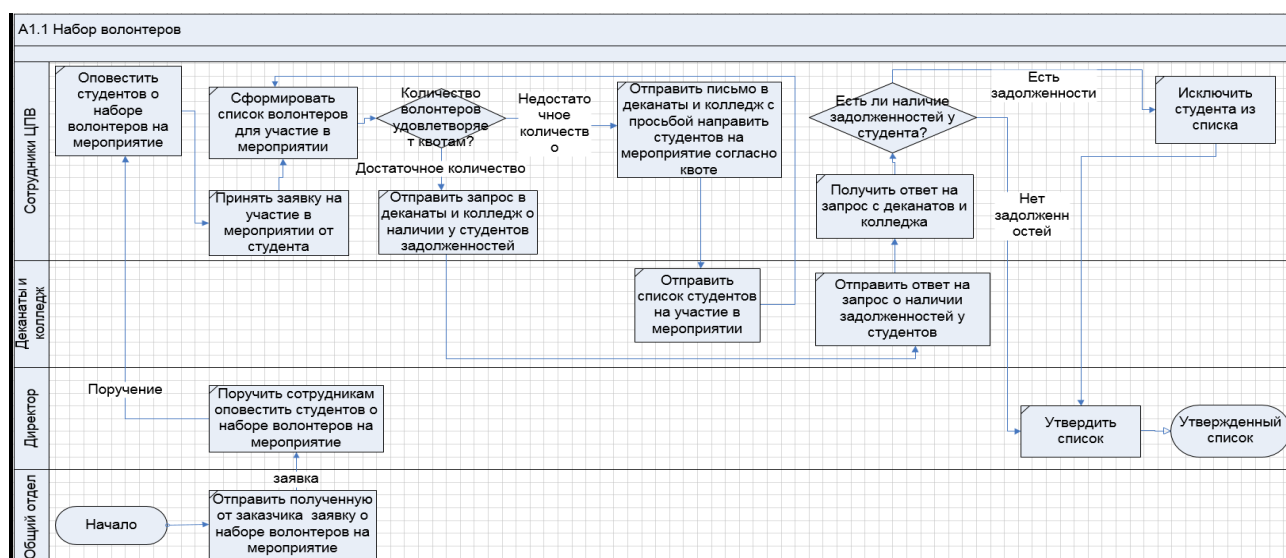


Рис.1. Бизнес-процесс «Набор волонтеров»

Поскольку ЦПВ ориентирован в основном на работу со студентами, то одной из основных задач, возникающей перед сотрудниками, является отслеживание успеваемости студентов. В данном бизнес-процессе такая возможность предусмотрена. После формирования списка студентов на участие в мероприятии, он высылается в деканаты факультетов с запросом об успеваемости каждого студента. В конечном итоге, сотрудники ЦПВ исключают из списка тех студентов, у которых уровень успеваемости ниже требуемого уровня.

Бизнес-процесс «Работа с волонтерами (утвержденные позиции)» является ключевым в работе ЦПВ так как управляет показателем «Процент волонтеров, отказавшихся от мероприятия» (рис. 2).

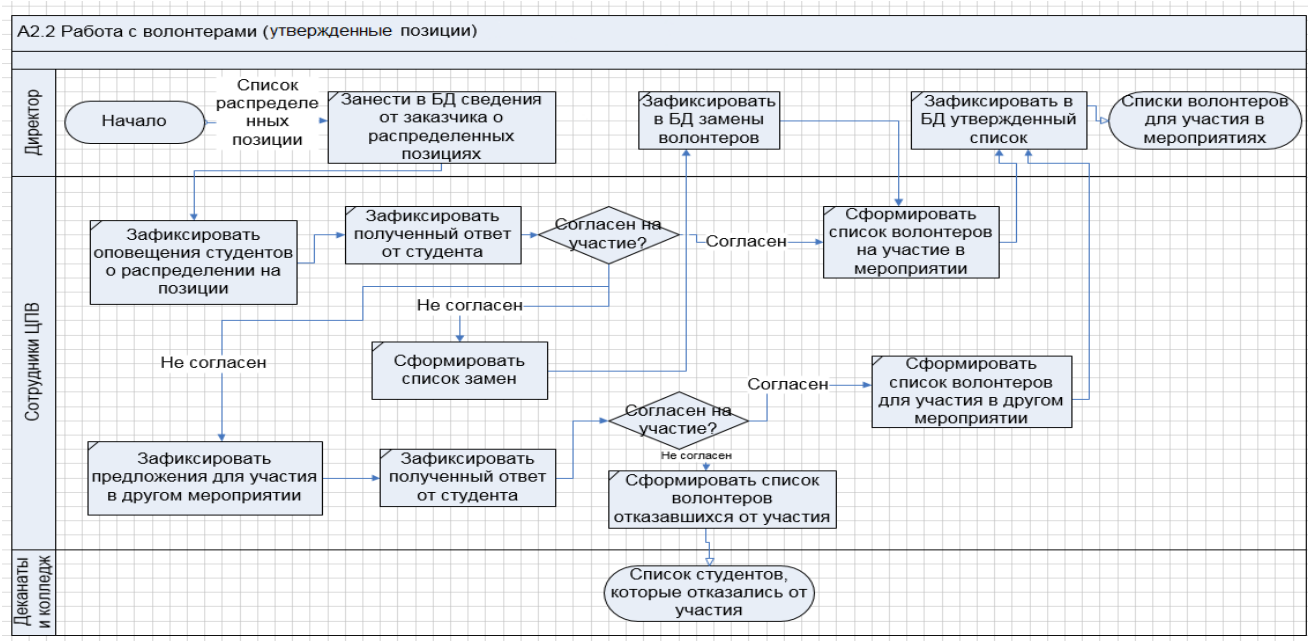


Рис. 2. Бизнес-процесс «Работа с волонтерами (утвержденные позиции)»

Задачами бизнес-процесса является отслеживание отказов от выхода на позиции, что в конечном итоге значительно улучшает организацию работы с волонтерами. Информационная система позволяет дополнительно повысить эффективность работы с волонтерами, которые по каким-то причинам отказались от участия в предыдущем мероприятии, за счет предложения участвовать в других мероприятиях. Система обеспечивает на основе архивных данных быстрый подбор волонтеров, имеющих необходимую подготовку для данной позиции, взамен отказавшихся от участия в мероприятии.

Диаграммы бизнес-процессов реализованы в среде пакета “Business Studio”, который позволяет на основе сведений о характеристиках функций процессов в автоматическом режиме сгенерировать регламентную документацию для сотрудников ЦПВ: должностные инструкции, регламент процессов и положение о центре.

Коррекция бизнес-процессов не требует навыков программирования, подготовленный сотрудник ЦПВ в состоянии самостоятельно внести в модель процесса необходимые изменения. Эта возможность позволяет постоянно поддерживать бизнес-процессы в актуальном состоянии и, следовательно, обеспечивать сотрудников актуальной регламентной документацией.

Кроме указанных выше, информационная система ЦПВ выполняет следующие функции:

- подбор волонтеров на мероприятие на основе архива волонтерской деятельности;
- формирование проекта распоряжения на освобождение от занятий на время обучения и на время проведения мероприятия;
- формирование ежемесячного отчета о проведенных мероприятиях;
- формирование ежемесячного отчета о количестве отработанных часов каждым волонтером.

Результаты работы будут использованы при автоматизации системы менеджмента качества Сочинского государственного университета.

УДК 004.934.8'1; ГРНТИ 28.23.29

ШУМОУСТОЙЧИВОСТЬ СИСТЕМ ИДЕНТИФИКАЦИИ ДИКТОРА

Д. Н. Кручок

*Белорусский государственный университет информатики и радиоэлектроники,
Республика Беларусь, Минск, den.dj.dnk@gmail.com*

Аннотация. В данной работе описан метод построения речевой маски для зашумленного речевого выражения с целью его использования в системах идентификации диктора. Приводится алгоритм получения мел-кепстральных коэффициентов с использованием речевой маски.

Ключевые слова. Речевая маска, мел-кепстральные коэффициенты шумоустойчивость, идентификация диктора

ROBUSTNESS TO NOISE OF SPEAKER IDENTIFICATION SYSTEMS

D. N. Kruchok

*Belarusian state university of informatics and radioelectronics,
Republic of Belarus, Minsk, den.dj.dnk@gmail.com*

Abstract. This paper describes a method of constructing a speech mask for a noisy speech utterance for speaker identification systems in acoustic noises. An algorithm for obtaining MFCC coefficients using speech mask is given.

Keywords. Speech mask, MFCC, robustness to noise, speaker identification

Системы идентификации диктора используются в разных акустических обстановках, и обычно разрабатываются с учетом заранее известной информации об окружающих шумах. Параметры такой системы корректируются и настраиваются под конкретную акустическую обстановку, пока не будет достигнута необходимая точность идентификации. Однако использование уже настроенной системы распознавания в окружающей обстановке с другим типом шума или в изменяющейся акустической обстановке вызовет её деградацию.

Подходы, обеспечивающие шумоустойчивость систем идентификации диктора, подразделяются [1]:

- На основе используемого характеристического вектора – шумовая составляющая убирается прямо из самих признаков диктора. Примерами таких методов являются кепстральная нормализация средним (англ. CMN – Cepstral Mean Normalization), фильтрация в модуляционной области (англ. RASTA – Relative Spectral Analysis) и др.

- На основе оценки используемого признака – подходы, которые меняют сами показатели оценки признаков на уровне рассматриваемого кадра или всего речевого выражения.

- На основе используемой модели диктора или решающего правила – комбинация нескольких решающих правил, или включение в модель диктора также и характеристических признаков шума и искажений для достижения надёжности.

Эти подходы ограничены при использовании в системах идентификации диктора в условиях с переменными акустическими шумами, так как требуют конкретной информации о типе шума (например, шум внутри автомобиля). Для случаев с переменной акустической обстановкой необходимо использовать методы, способные компенсировать аддитивные искажения произвольного типа. Они основаны на наблюдении, что речевой сигнал имеет высокую степень избыточности, когда информация об основных речевых характеристиках присутствует даже в очень искаженных или зашумлённых речевых выражениях (англ. missing data approaches [1]). Суть таких методов заключается в том, что речевой сигнал в частотной области маркируется на каждой отдельной частоте по наличию признака речи или шума – образуется речевая маска сигнала в частотной области, с помощью которой можно восстановить сигнал во временной области без шумовой составляющей.

Пусть $s(n)$ – дискретизированный сигнал во временной области ($n = 0, 1, \dots, N$; N – количество отсчётов в сигнале), который содержит только речь без шума, а $S(t, f)$ –

прологарифмированное значение спектрограммы сигнала в кадре с номером t , на частоте f . На рисунке 1 представлена схема получения $S(t, f)$.

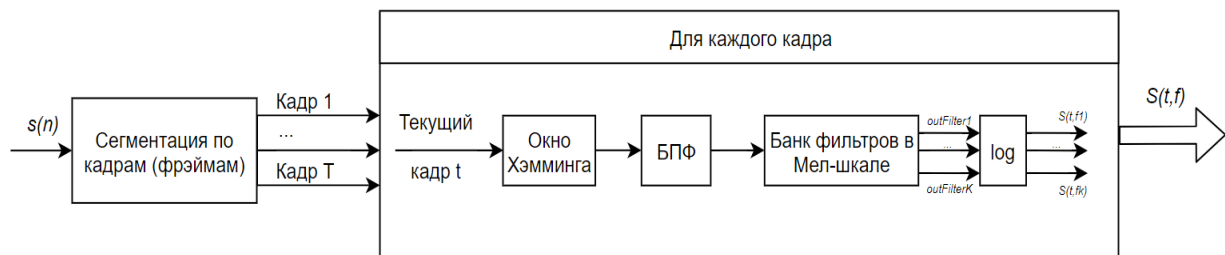


Рис 1. Схема получения спектрограммы речевого сигнала $S(t, f)$

Уровень зашумления для кадра t , на частоте f количественно можно найти по отношению сигнал-шум (англ. SNR – signal to noise ratio):

$$SNR(t, f) = 10 \cdot \log_{10} \frac{S_p(t, f)}{N_p(t, f)}, \quad (1)$$

где $S_p(t, f)$ – спектральная плотность мощности (СПМ) чистого речевого сигнала;
 $N_p(t, f)$ – СПМ шума.

Таким образом, для каждого кадра и частоты большое значение SNR свидетельствует о том, что эта область «надёжная» или речевая, и наоборот, маленькое значение соответствует зашумлённой области. Для построения маски, которая будет иметь такую же размерность, что и $S(t, f)$, каждой области (с определёнными значениями t и f) будет ставиться значение 1 или 0, в зависимости от уровня SNR и выбранного порога θ надёжности (в дБ): $m(t, f) = 1$, при $SNR(t, f) > \theta$ и $m(t, f) = 0$ – для остальных случаев. Порог надёжности не всегда постоянен и обычно выбирается в интервале $[-3\text{дБ}; 3\text{дБ}]$ [2]. Речевая маска может быть и не бинарной и своими значениями представлять вероятность речи [3].

На рисунке 2 изображены спектрограммы чистого речевого сигнала и зашумлённого. Речевой сигнал содержит фразу: «штурман просил продолжать разворот», частота дискретизации равна 16 кГц, тип используемого шума – babble, уровень зашумления SNR составляет минус 5 дБ.

Полученные речевые маски с различным параметром θ представлены на рисунке 3. Чем больше порог, тем больше данных не будут учитываться в дальнейшей обработке.

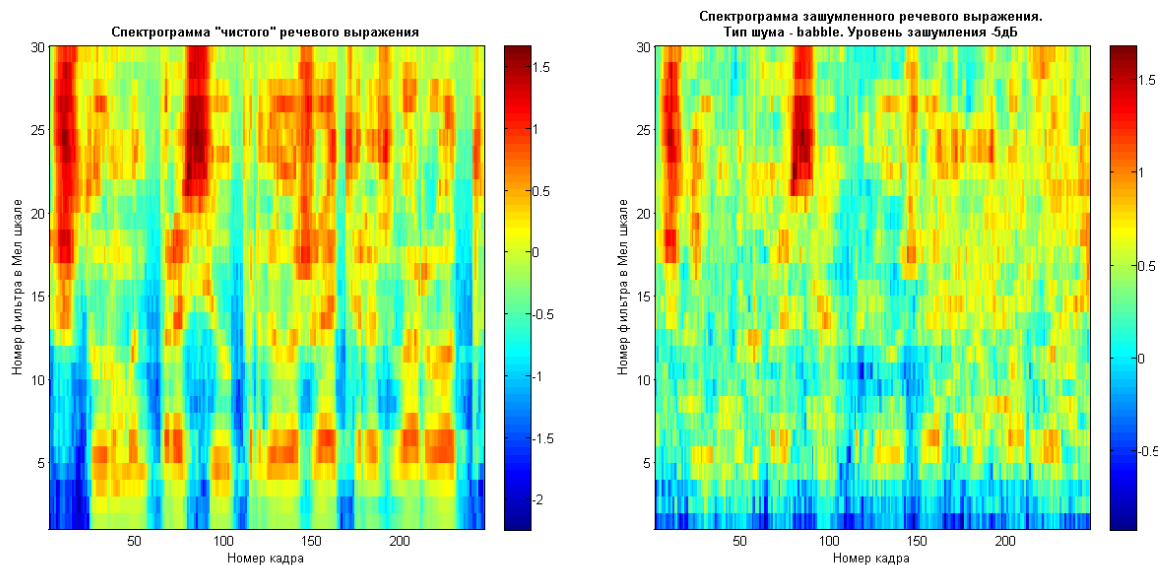


Рис.2. Спектрограммы «чистого» речевого сигнала (слева) и зашумленного шумом babble с уровнем SNR= -5дБ (справа)

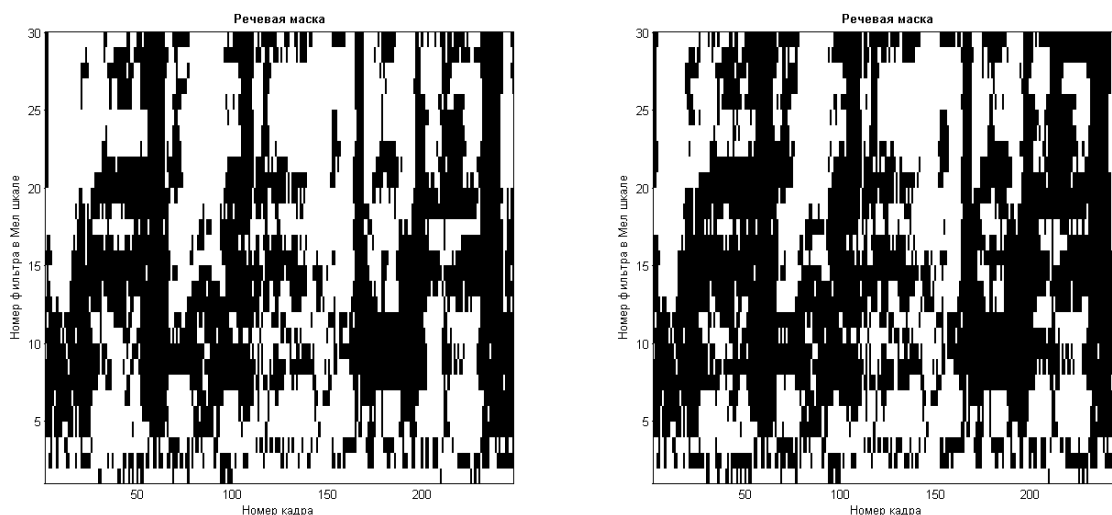


Рис.3. Полученная речевая маска с порогом надёжности $\theta = 0$ дБ (слева) и $\theta = 3$ дБ (справа)

Для построения идеальной речевой маски необходимо иметь данные о «чистом» речевом сигнале, но при идентификации диктора в акустических шумах имеется только зашумлённый речевой сигнал. Для использования данного подхода в идентификации диктора при изменчивых акустических условиях необходимо оценивать эти речевые маски. Подходы, с помощью которых их можно оценить, используют свойства речевого сигнала и подразделяются на [1]:

- методы, основанные на оценке отношения сигнал-шум;
- методы, основанные на перцептуальном и слуховом восприятии (антропоморфические методы);
- методы, основанные на классификации.

Рассмотрим метод построения речевой маски на основе оценки SNR. Для расчёта SNR используется среднее значение спектральной плотности мощности шума, взятого из речевого выражения из первых нескольких кадров, когда ещё нет энергии речи. Тогда на основе за-

шумленного речевого сигнала можно получить СПМ «чистого» речевого сигнала следующим образом [4,1]:

$$\hat{S}_P(t, f) = \begin{cases} X_P(t, f) - \hat{N}_P(t, f), & X_P(t, f) - \hat{N}_P(t, f) > \gamma X_P(t, f), \\ \gamma X_P(t, f) & \text{в остальных случаях} \end{cases} \quad (2)$$

где

$$\hat{N}_P(t, f) = \frac{1}{T_{avg}} \sum_{\tau=1}^{T_{avg}} X_P(\tau, f) - \text{спектр мощности для первых } T \text{ кадров;}$$

γ – малый масштабирующий коэффициент ($\gamma \approx 0$), необходим, чтобы значения спектра мощности были неотрицательными.

Оценка речевой маски будет происходить также как и в предыдущем случае, только вместо заранее известного «чистого» сигнала, необходимо использовать полученный.

Приведём алгоритм варианта получения характеристического вектора на основе мел-кепстральных коэффициентов с использованием речевой маски:

Шаг 1. Зашумлённый речевой сигнал $s[n]$ разбивается на кадры с определённой длиной (обычно выбирается между 20 и 30 мс [1]) и перекрытием.

Шаг 2. Для каждого кадра выполняется:

Шаг 2.1. Все отсчёты текущего кадра пропускаются через оконную функцию (например, окно Хэмминга).

Шаг 2.2. Берем амплитуды половины значений быстрого преобразования Фурье (БПФ) (например, если длина БПФ была 512 отсчётов, то для дальнейшей обработки берётся 256 отсчётов).

Шаг 2.3. Пропускаем полученные значения через банк фильтров в шкале Мел и логарифмируем их – получаем для каждого кадра мощность на выходах каждого фильтра.

Шаг 2.4. Сохраняем полученные значения текущего кадра в общую матрицу.

Шаг 3. Находим спектр мощности шума на основе первых T кадров зашумленного выражения по формуле 2.

Шаг 4. По формуле 1 оцениваем параметр SNR для каждого выхода фильтра для каждого кадра.

Шаг 5. Сравниваем полученные SNR с порогом надежности, и для тех кадра и номера фильтра ставим в соответствие значение 0 или 1 – получаем речевую маску.

Шаг 6. По полученной маске получаем спектрограмму только речевого сигнала.

Шаг 7. Берем дискретное косинусное преобразование (ДКП) от полученной спектрограммы «чистого» сигнала и получаем кепстральные коэффициенты.

Таким образом, получены кепстральные коэффициенты на основе речевой маски, что увеличит показатели распознавания диктора в сравнении с обычной процедурой получения мел-кепстральных коэффициентов.

Библиографический список

1. Togneri R., Pullella D. An Overview of Speaker Identification Accuracy and robustness Issues // IEEE Circuits and systems magazine. 2011. P. 23–58.
2. B. Raj and R. M. Stern, «Missing-feature approaches in speech recognition», IEEE Signal Processing Mag., vol. 22, no. 5, pp. 101–116, 2005.
3. A. Morris, J. Barker, and H. Bourlard, «From missing data to maybe useful data: Soft data modeling for noise robust asr», Stratfordupon-Avon, U.K., Tech. Rep., 2001.
4. S. Boll, «Suppression of acoustic noise in speech using spectral subtraction», IEEE Trans. Acoust. Speech Signal Process., vol. 27, no. 2, pp. 113–120, 1979.

УДК 004.67; ГРНТИ 50.41.23

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ СТАТИСТИЧЕСКОГО АНАЛИЗА ДАННЫХ С РАСШИРЯЕМОЙ АРХИТЕКТУРОЙ НА ПЛАТФОРМЕ ASP.NET MVC

Р.В. Ерхов*Рязанский государственный радиотехнический университет,
Россия, Рязань, romanerkhov@yandex.ru*

Аннотация. В данной работе рассматривается веб-приложение, которое включает в себя некоторые методы статистического анализа данных. Рассматривается расширяемая архитектура веб-приложения на платформе ASP.NET MVC, которая поддерживает добавление в проект сторонних сборок.

Ключевые слова. Веб-приложение, методы статистического анализа данных, расширяемая архитектура, ASP.NET MVC

PROGRAMMATIC IMPLEMENTATION OF THE STATISTICAL DATA ANALYSIS WEB APPLICATION WITH EXTENSIBLE ARCHITECTURE ON THE ASP.NET MVC PLATFORM

R.V. Erkhov*Ryazan State Radioengineering University,
Russia, Ryazan, romanerkhov@yandex.ru*

Abstract. This paper discusses a web application that includes some methods of statistical data analysis. Discusses the extensible architecture of a web application on the ASP.NET MVC platform that supports adding third-party assemblies to a project.

Keywords. Web application, statistical data analysis methods, extensible architecture, ASP.NET MVC.

Введение

В современном мире большими темпами развивается Интернет. Это развитие может выражаться и представлять собой различные бизнес проекты, разнообразные сайты, различные сервисы и прочие ресурсы сети Интернет. В связи с этим развитием ресурсов также увеличивается объем информации в сети, как различных компаний, так и частных лиц у которых существуют какие либо сервисы или сайты. Также большинству компаний и организаций необходимо вести прогнозирование, по какому либо виду учета и статистики для увеличения финансовых показателей так и для автоматизации и корректировки бизнес-процессов. У большинства сайтов с огромным количеством информации о посещениях, об используемом трафике, количестве переходов на страницы и т.д. может также возникнуть необходимость в учете статистики.

Для улучшения ресурсов сети Интернет зачастую требуется автоматизировать большой объем информации, а также учитывать ее анализ и статистику [1-2]. Для учета и анализа данных статистики могут применяться и эффективно использоваться методы статистического анализа данных. Необходимость своевременной корректировки бизнес-процессов и анализа данных выражается в использовании методов статистического анализа данных в режиме реального времени.

Цель разработки

Таким образом, целью разработки является реализация веб-приложения, которое будет включать в себя некоторые методы статистического анализа данных. Необходимо реализовать расширяемую архитектуру веб-приложения, которая будет поддерживать добавление к приложению сторонних сборок. Сторонние сборки предполагают дополнить веб-

приложение необходимыми методами статистического анализа данных. Реализацию веб-приложения необходимо произвести с помощью платформы ASP.NET MVC. Платформа ASP.NET MVC основывается на архитектурном шаблоне “модель-представление-контроллер”, что предоставляет возможности расширяемости архитектуры и более гибкого тестирования кода [3]. Платформа ASP.NET MVC используется в комбинации со всеми преимуществами платформы ASP.NET [4].

Структура веб-приложения

Наглядная структура веб-приложения представлена на рисунке 1.

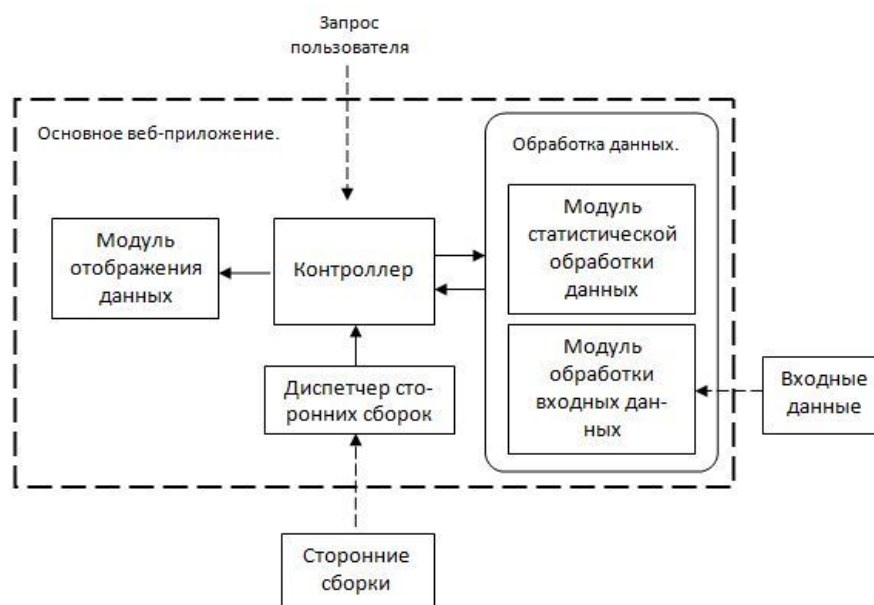


Рис. 1. Структура веб-приложения статистического анализа данных

На основе представленной структуры веб-приложения можно рассмотреть абстрактную функциональность приложения. Основное веб-приложение можно представить в виде блока обработки данных, контроллера и модуля отображения данных.

Блок обработки данных представляет собой модуль статистической обработки данных и модуль обработки входных данных. В свою очередь модуль статистической обработки данных реализует основные методы статистического анализа данных основного веб-приложения. Модуль обработки входных данных отвечает за загрузку входных данных на сервер, удаление входных данных, получение входных данных с сервера, обработку входных данных и их импорт в приложение. Входные данные представляют собой текстовые файлы со значениями какой-либо статистики.

Контроллер отвечает за обработку запросов пользователей, вызов нужных операций из модулей блока обработки данных и генерацию необходимых отображений на поступившие запросы пользователей [5]. Контроллер также отвечает за обработку и взаимодействие с блоком диспетчера сторонних сборок. Диспетчер сторонних сборок реализует проверку наличия сборок и их обработку. Обработка в свою очередь включает в себя извлечение из сборок необходимых операций и генерацию списка отображений. Также диспетчер сторонних сборок отвечает за регистрацию ссылок на сборки в основном веб-приложении.

Сторонние сборки представляют собой реализацию дополнительных методов статистического анализа данных. Сборки представляют собой файлы с расширением dll, их структура будет рассматриваться далее.

Модуль отображения данных реализует отображение сгенерированных веб страниц. Данный модуль отображает как отображения сгенерированные контроллером, так и отображения сгенерированные диспетчером сторонних сборок. Модуль отображения данных также воспринимает от диспетчера сторонних сборок информацию о названиях дополнительных методов статистического анализа данных и реализует базовое отображение всех доступных методов анализа для основного веб-приложения.

Представленная архитектура, реализованная таким образом, позволяет в режиме реального времени без перезапуска и перекомпиляции проекта добавлять в него дополнительные методы, которые реализуют необходимые операции анализа данных.

Структура сторонней сборки

Наглядная структура сторонней сборки представлена на рис. 2.

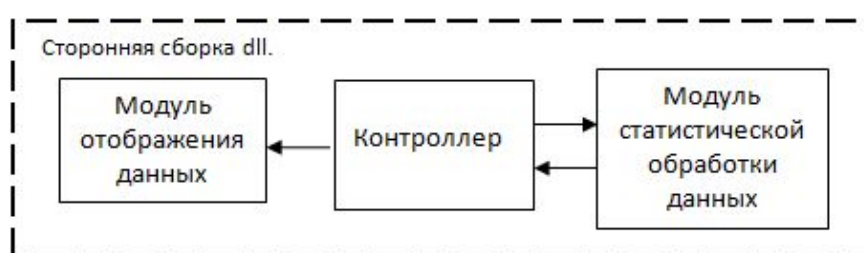


Рис. 2. Структура сторонней сборки

На основе представленной структуры сторонней сборки можно рассмотреть абстрактную функциональность сборки. Сборку можно представить в виде модуля статистической обработки данных, контроллера и модуля отображения данных.

Модуль статистической обработки данных реализует дополнительный метод статистического анализа данных сторонней сборки.

Контроллер отвечает за вызов метода анализа данных из модуля статистической обработки данных и генерацию отображения для реализованного метода анализа в данной сборке. Контроллеру сторонней сборки необходимо содержать название метода анализа данных для того, чтобы модуль отображения данных основного веб-приложения воспринял от диспетчера сторонних сборок информацию о названии дополнительного метода статистического анализа данных. Еще контроллеру необходимо содержать название маршрута собственного метода действия для того чтобы модуль отображения данных основного веб-приложения воспринял от диспетчера сторонних сборок информацию о названии маршрута и совершил переход к нужному методу действия контроллера по названию маршрута. При генерации собственного отображения контроллеру необходимо прикрепить базовое отображение основного веб-приложения для того чтобы модуль отображения данных основного веб-приложения реализовал базовое отображение всех доступных методов анализа данных.

Модуль отображения данных сторонней сборки реализует отображение сгенерированное контроллером.

Представленная архитектура сторонней сборки, реализованная таким образом, позволяет в режиме реального времени без перезапуска и перекомпиляции проекта добавлять в него дополнительные методы, которые реализуют необходимые операции анализа данных. Также единая спецификация для проектирования сторонних сборок позволяет разработчикам реализовывать дополнительные методы статистического анализа данных в рамках шаблона MVC.

Основные возможности

Рассматриваемое веб-приложение статистического анализа данных включает в себя следующие возможности:

- загрузка файлов статистики на сервер и возможность их удаления;
- просмотр содержимого загруженных файлов;
- скачивание загруженных файлов с сервера;
- извлечение, обработка и импорт загруженных файлов в расчеты веб-приложения;
- расчет среднего значения данных статистики;
- проведение анализа корреляции данных статистики;
- проведение анализа парной линейной регрессии;
- добавление в проект дополнительных методов статистического анализа данных в режиме реального времени;
- возможность для разработчиков реализовывать дополнительные методы в рамках шаблона MVC.

Заключение

В результате было разработано веб-приложение, которое включает в себя некоторые методы статистического анализа данных. Была реализована расширяемая архитектура веб-приложения поддерживающая добавление к приложению сторонних сборок, которые в свою очередь содержат дополнительные методы статистического анализа данных. Была достигнута единая спецификация для проектирования сторонних сборок, что позволило разработчикам реализовывать дополнительные методы в рамках шаблона MVC.

Библиографический список

1. Пруцков А.В., Цыбулько Д.М. Теоретико-множественное представление метода обработки количественных числительных естественных языков и особенности их перевода в различных странах // Вестник Рязанского государственного радиотехнического университета. 2014. № 50-2. С. 69-75.
2. Prutzkow A. Interlingua-based numeral translation in Web-application with knowledge-testing. The European Proceedings of Social & Behavioural Sciences EPSBS «The European Proceedings of Social & Behavioural Sciences EPSBS» 2017. pp. 290-298.
3. Джесс Чедвик, Хришикеш Панда, Тодд Снайдер. ASP.NET MVC 4: разработка реальных веб-приложений с помощью ASP.NET MVC.: Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2013. – 432 с.: ил. – Парал. тит. англ. ISBN 978-5-8459-1841-3 (рус.)
4. Ерхов Р.В. Методы разработки и архитектура современных веб-приложений.: Новые информационные технологии в научных исследованиях: материалы XXI Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязань: РГРТУ, 2016. – С. 129-131.
5. Архитектура ASP.NET MVC 5 [Электронный ресурс], URL: https://professorweb.ru/my/ASP_NET/mvc/level1/1_7.php (дата обращения 13.02.18).

УДК 004.67; ГРНТИ 50.41.23

ПРОГРАММНАЯ СИСТЕМА ПРОВЕРКИ СОВПАДЕНИЯ ТЕКСТОВ ПРОГРАММ

Н.Н. Степанов

Рязанский государственный радиотехнический университет

Россия, Рязань, nikita.stepanov150@gmail.com

Аннотация. В данной работе рассматриваются средства и различные алгоритмы поиска совпадения текстов программ.

Ключевые слова. Антиплагиат, статический анализатор, автоматическая проверка, повышение качества кода, лабораторные работы.

PROGRAM SYSTEM OF CHECKING THE COMPARISON OF THE TEXTS OF PROGRAMS

Ryazan State Radioengineering University

Russia, Ryazan, nikita.stepanov150@gmail.com

Abstract. In this article, we consider tools and various algorithms for finding the coincidence of program texts.

Keywords. Antiplagiarism, static analyzer, automatic check, code quality improvement, laboratory work.

На текущий момент сфера информационных технологий быстро развивается. В высших учебных заведениях на технических направлениях изучаются языки программирования. Для проверки знаний и отработки навыков, практики, они выполняют лабораторные работы. Нередко можно встретить похожие работы. Для контролирования их выполнения необходимо разработать автоматизированную систему проверки.

Большое количество разработчиков разных уровней усложняет задачу. Для этого придумали разные статические анализаторы, которые предназначены для поиска уязвимостей. Эти системы встроены в промышленную разработку, их запуск производится каждый раз непосредственно при загрузке на сервер.

Внимательная проверка занимает довольно много времени: проект обычно состоит из большого количества файлов. В итоге получается довольно трудоемкий и длительный процесс.

Целью данной работы является разработка программной системы поиска совпадений исходных кодов программ, которая должна иметь следующий функционал:

- Использовать набор проектов в качестве исходных данных;
- Очистить данные перед проверкой;
- Провести посимвольную проверку совпадения;
- Провести логический анализ работ;
- Выдать результат в удобной форме;
- Иметь графический интерфейс.

Программная система должна иметь графический интерфейс, где возможно указать файлы (проекты) для проверки. Перед этим необходимо подготовить файлы. Делается это следующим образом: очищаем каждый файл от незначущей информации, которая может повлиять на исходный результат работы:

1. Удалить метаинформацию об импортированных библиотеках, классах, ведь в данном анализе они не играют никакой роли;
2. Удалить все комментарии, так как их можно использовать в любых местах, что может повлиять на результаты во время посимвольной проверки;
3. Удаление документации (перед методами, классами), аналогично пункту 2;
4. Отсортировать поля, чтобы их порядок не отражался на результатах;
5. Отсортировать методы, аналогично пункту 4.

Далее проекты подвергаются полной посимвольной проверке. На основе результатов подсчитывается уникальность документа, создается отчет, который отображает заимствованные участки кода, система пытается выяснить какой файл, предположительно, был первоисточником, путем анализа заимствованных участков.

Сравним исходные коды программ и убедимся в целесообразности всего вышесказанного. Допустим, у нас есть такой метод:

```
public static String readFile(String path) throws IOException {
    BufferedReader br = new BufferedReader(new FileReader(path));
    try {
        // create new string
        StringBuilder sb = new StringBuilder();
        // buffer
        String line;
        // while line != null
        while ((line = br.readLine()) != null) {
            sb.append(line).append(LINE_SEPARATOR);
        }
        return sb.toString();
    } finally {
        // close stream
        br.close();
    }
}
```

При сравнении данного фрагмента кода и аналогичного, но без комментариев, совпадение равно 65%. Данный результат неверный, ведь они полностью одинаковые. При обработке его должным образом, без комментариев, ведь они не влияют на работу программы совпадение равно 100%.

Рассмотрим еще одну проблему. У нас имеется класс с двумя полями:

```
public class Runner {
    private int y1 = 3971;
    private int y2 = 1013;
    ...
}
```

При перестановке местами двух полей получаем 0% совпадения. Что еще раз доказывает необходимость обработки перед проверкой.

Рассмотрим еще раз метод `readFile`, только переименуем все переменные

```
public static String readFile(String filePath) throws IOException {
    BufferedReader bufferedReader = new BufferedReader(new
    FileReader(filePath));
    try {
        StringBuilder stringBuilder = new StringBuilder();
        String buffer;

        while ((buffer = bufferedReader.readLine()) != null) {
            stringBuilder.append(buffer).append(LINE_SEPARATOR);
        }
    }
}
```

```
    }  
    return stringBuilder.toString();  
} finally {  
    bufferedReader.close();  
}  
}
```

При простом переименовании переменных эти два метода, при простом анализе, будут абсолютно разными. Необходимо учитывать ключевые особенности программного кода:

- Алгоритм должен распознавать основные ключевые слова языка;
- Необходимо не учитывать имена переменных;
- Существуют такие ситуации, когда при перестановке строк код не должен распознаваться как совершенно другой;
- Необходимо игнорировать комментарии, пробельные символы и переводы строк.

Поэтому, если данный способ не выявил похожие работы, тогда на основе каждого проекта будет строиться абстрактное синтаксическое дерево, которое позволит проанализировать проекты без учета имен переменных, методов. С его помощью можно проанализировать логику программы. На основе его работы тоже будет строиться отчет.

Все сделано для повышения надежности распознавания. Также, генерация абстрактного синтаксического дерева расходует ресурсы в десятки, а то и в сотни раз больше, нежели первый вариант. Приходится искать компромисс для повышения скорости, увеличения производительности работы нашей системы.

Программная система состоит из нескольких модулей: модуль взаимодействия пользователем (интерфейс), модуль загрузки файлов из zip архива, модуль чтения файлов из папки, модуль очистки исходного кода, модуль поиска совпадения, модуль генерации отчета. Все модули не связаны между собой, что позволяет легко добавлять и удалять функционал.

Данная программная система будет внедрена в дисциплину «Программирование на Java» [1-2], преподаваемую на кафедре «Вычислительная и прикладная математика» Рязанского государственного радиотехнического университета. Она поможет преподавателю проверять работы студентов. В дальнейшем возможно расширение функционала данной системы для использования в разных дисциплинах. Программная система будет разработана на языке Java [3-4] в рамках выпускной квалификационной работы.

Библиографический список

1. Пруцков А.В. Особенности преподавания промышленной разработки программных продуктов в технических вузах // Современные технологии в науке и образовании – СТНО-2017: материалы 2-й Междунар. науч.-техн. и науч.-метод. конф. / под общ. ред. О.В. Миловзорова / Рязан. гос. радиотехн. ун-т. – Рязань, 2017. – С. 40-41.
2. Пруцков А.В. Программирование на языке Java. Введение в курс с примерами и практическими заданиями: учебник. – М.: КУРС, 2018. – 208 с.
3. Блинов И.Н., Романчик В.С. Java. Методы программирования: учеб.-метод. пособие. – Минск:Изд-во «Четыре четверти», 2013. – 896 с.
4. Брюс Эккель. Думаем в Java: учебное пособие. – Изд-во «Prentice Hall Ptr», 2006. – 1095 с.

УДК 519.872; ГРНТИ 50.05.19

ВИЗУАЛЬНАЯ ПРОГРАММНАЯ СИСТЕМА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

А.И. Мартышкин

*Пензенский государственный технологический университет,
Россия, Пенза, Alexey314@yandex.ru*

Аннотация. В статье описана визуальная программная система имитационного моделирования вычислительных систем. Предложенная система имеет возможность проводить имитационное моделирование и получать характеристики всей вычислительной системы, а также отдельных ее блоков. Основным отличием созданной программы от аналогичных является возможность исследования вычислительной системы, представленной сетью массового обслуживания, которая состоит из систем с ограничением длины очереди перед обрабатывающим прибором и приоритетных систем массового обслуживания, что позволяет снимать информацию по более широкому спектру исследуемых данных. Программная система довольно проста в эксплуатации и не требует дополнительных знаний. Программа может применяться создателями новых операционных систем для их анализа, исследования и отладки на начальном этапе проектирования.

Ключевые слова. Диспетчер задач, дисциплина обслуживания, многопроцессорная система, комплекс программ, система массового обслуживания, имитационное моделирование

VISUAL PROGRAM SYSTEM OF IMITATION MODELING OF COMPUTER SYSTEMS

A.I. Martyshkin

*Penza State Technological University,
Russia, Penza, Alexey314@yandex.ru*

Abstract. The article describes the visual software system of simulation of computer systems. The proposed system has the ability to carry out simulation modeling and obtain characteristics of the entire computer system, as well as its individual blocks. The main difference between the created program and similar is the ability to study the computer system provided by the Queuing network, which consists of systems with a limit of the queue length before the processing device and priority Queuing systems, which allows you to capture information on a wider range of the studied data. The software system is easy to operate and requires no additional knowledge. The program can be used by creators of new operating systems for their analysis, research and debugging at the initial stage of design.

Keywords. Task manager, service discipline, multi-processor system, a complex pro-gram, queuing system, simulation

При разработке и совершенствовании любых современных вычислительных систем присутствует необходимость исследования их характеристик. Создание устройства физически требует больших трат времени и материальных ресурсов, что весьма нежелательно при частом изменении структуры системы. Принимая во внимание этот факт для исследования характеристик вычислительных систем часто бывают применены программные модели, которые не являются физическими устройствами, а описаны определенным способом в памяти ЭВМ. Как следствие, ее создание и совершенствование (изменение, модификация) производится без финансовых затрат. Исследование вычислительных систем с применением моделирования является важной составляющей этапа системного проектирования [1]. По полученным характеристикам нескольких вариантов моделей можно сравнивать различные вычислительные системы, проводить анализ их производительности при вариации нагрузки и т.д. Более того не последнюю роль играет познавательная ценность моделирования (исследования). Как правило, это обстоятельство и приводит к массовому применению моделирования в учебном процессе. В статье описывается программа для имитационного моделирования вычислительных систем, которая отличается дружественным интерфейсом пользователя, что приводит к упрощению процедуры задания параметров объекта моделирования и получению большего числа экспериментальных данных за единицу времени, в отличие от существующих программ-аналогов.

Описываемая в работе программная система при создании модели содержит в себе следующие элементы: источник заявок, очередь, обслуживающее устройство, терминатор заявок. Кроме того программа имеет поддержку возможности определения вероятностей передач заявок от одного элемента (узла) модели к другому, а также редактирования параметров элементов и создания отчёта о результатах проведенного исследования. В качестве моделей источников заявок применены источники с временным интервалом между поступлением заявок, распределенным по экспоненциальному, равномерному и постоянному закону, а также законам Пуассона и Гаусса. Описаны модели приоритетных и беспriorитетных очередей с дисциплинами обслуживания FIFO и LIFO, модели одноканальных и многоканальных блоков обслуживания с распределенным по различным законам временем обслуживания.

Сегодня известны программные продукты, функционал которых способен выполнять имитационное моделирование вычислительных систем. Эти программы, как правило, работают в текстовом режиме. Все существующие программные средства для имитационного моделирования систем имеют сходный функционал и пользовательский интерфейс. В них модель исследуемой системы задается путем текстового описания, состоящего из последовательности блоков, описывающих элементы модели (очередь, устройство, и т.д.) и переходы между ними, а уже сами результаты расчета записываются в текстовый файл. Также существует инструментарий для визуального имитационного моделирования, разработанный для систем автоматики и мало приспособленный для исследования сложных процессов в вычислительных системах.

Предлагаемая программа обладает визуальным интерфейсом, который позволяет наглядно отображать и изменять модель в ходе проведения исследований и получения экспериментальных данных [2]. Наряду с этим модель сохраняется на диске в виде текстового файла – полного и подробного описания модели. Этот факт предполагает универсальность использования модели, возможность увеличения функционала программы без изменения формата представления исследуемой модели, расчет которой производится непрерывно до выполнения условий: генерации строго определенного количества заявок или прошествия некоторой величины модельного времени. Результатом исследования, полученным в ходе выполнения экспериментов, является созданный программой HTML-файл, содержащий основные полученные характеристики. Кроме того, существует возможность получить подробное описание всего процесса моделирования в текстовом виде (подробный путь следования заявок по системе в виде протокола). Модель задается либо с помощью ввода текста описания в необходимое поле, либо с помощью визуального интерфейса, тогда модель представляется в виде схемы из условных обозначений устройств, очередей и генераторов на экране ПЭВМ (рис.1), где также размещаются возможные переходы заявок между элементами модели. Максимальное количество определенных в модели элементов может быть выбрано исходя из объема оперативной памяти ПЭВМ, т.к. наряду со статически выделенной памятью под параметры элементов в процессе моделирования под каждую заявку динамически выделяется свободная память (принято, что максимальное число элементов 450) [3]. При соединении указывается вероятность переходов, причем программа автоматически не дает вводить вероятность больше допустимой (сумма вероятностей всех переходов из одного элемента должна быть равна единице). Параметры элементов можно ввести или отредактировать после двойного щелчка на их изображении, там же задаются дополнительные параметры (длина и тип очереди, закон распределения генератора заявок и т.д.). Созданные в программном продукте модели можно сохранять на любом диске, доступном ПЭВМ и загружать с любого диска в виде текстовых файлов.

Результаты моделирования сохраняются в HTML-формате в виде таблиц и выводятся на экран с помощью установленного по умолчанию web-браузера (рис. 2).

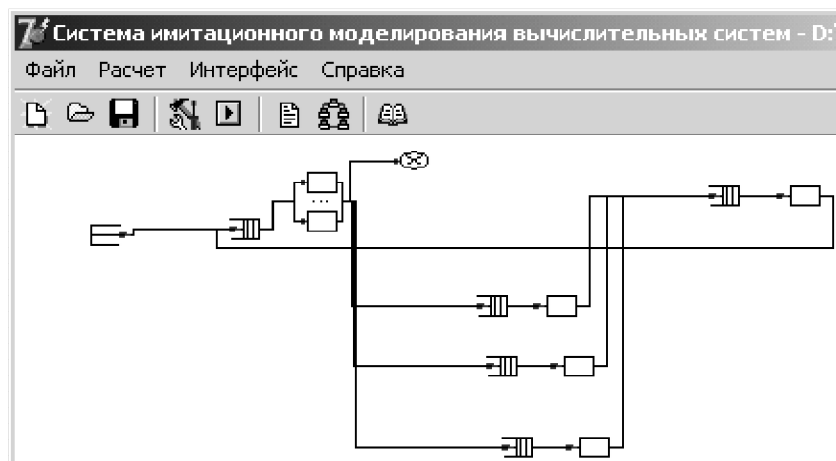


Рис. 1. Общий вид окна программы с построенной моделью

7	КПДП	1	0,1	Равномерное	0,0005	в ус-во №1 тип 1 с вер. 0,33 в ус-во №2 тип 1 с вер. 0,33 в ус-во №3 тип 1 с вер. 0,34
---	------	---	-----	-------------	--------	--

Рассчитанные данные:

Всего прошло заявок:	10000
Время моделирования:	10,0327914901718

Генераторы заявок:

Наименование	Обслуженных заявок	Необслуженных заявок	Среднее время пребывания заявок в системе	Минимальное время пребывания заявки в системе	Максимальное время пребывания заявки в системе
1. Источник_заявок	9856	144	0,0128932448472882	0,000804055535234438	0,146079190564979

Очереди:

Наименование	Максимальная длина очереди	Общее количество входов	Количество "нулевых" входов	Средняя длина очереди	Среднее время ожидания в очереди	Среднее время ожидания в очереди без учета "нулевых" входов	Максимальное время ожидания в очереди
1. ЦП/ОП1	7	6623	2504	0,469120584900387	0,000710643063876345	0,00114265331683732	0,00628894867920127
2. ЦП/ОП2	6	6528	2628	0,392522946922178	0,000603263002692707	0,00100976945681484	0,00577576170316751
3. ЦП/ОП3	8	7758	2178	0,706278101515519	0,000913372122528952	0,00126988188648378	0,00686059170668951
4. ВЗУ1	1	3363	2724	0,00404326569895388	1,20622187621077E-5	6,34823813723971E-5	0,00027581051997827
5. ВЗУ2	10	4468	500	2,92434897203811	0,00656655852304312	0,00739399785306349	0,0221006825645258
6. ВЗУ3	10	3078	249	3,45412197195253	0,0112587672274921	0,0122497297724353	0,0307993585499711
7. КПДП	2	10909	6697	0,124986769766504	0,000114947859574429	0,000297712773052568	0,000968932116590437

Рис. 2. Представление результатов расчета

Перед тем, как начать расчет, модель проверяется на корректность. Эту процедуру необходимо проводить, т.к. без контроля правильности возможна ситуация создания модели, где с течением времени будут скапливаться заявки. При генерировании очередной заявки под нее необходимо выделить память, а при ее уничтожении память нужно освободить. При бесконтрольном неограниченном наращивании числа заявок в исследуемой системе наступает переполнение памяти, затрагиваются некоторые области системного кэша, что зачастую может приводить к программному сбою.

Библиографический список

1. Бикташев Р.А., Мартышкин А.И., Востоков Н.Г. Комплекс программ для определения характеристик диспетчеров задач многопроцессорных систем с использованием приоритетных стохастических сетей массового обслуживания // Фундаментальные исследования. – 2013. – № 10 (ч. 1). – С. 13-20.
2. Архангельский А. Я. Программирование в Delphi 7. – М.: Изд-во Бином, 2003. – 1152 с.
3. Мартышкин А.И., Бикташев Р.А., Востоков Н.Г. Программный комплекс для имитационного моделирования диспетчеров задач многопроцессорных систем с использованием приоритетных сетей массового обслуживания // Фундаментальные исследования. – 2014. – № 11(ч. 10). – С. 2155-2159.

УДК 004.337; ГРНТИ 50.33.04

АНАЛИЗ МНОГОПРОЦЕССОРНОЙ СИСТЕМЫ С ОБЩЕЙ МНОГОМОДУЛЬНОЙ ПАМЯТЬЮ

А.И. Мартышкин

*Пензенский государственный технологический университет,
Россия, Пенза, Alexey314@yandex.ru*

Аннотация. В статье приводится структура многопроцессорной системы с общей многомодульной памятью. Показано, что в такой системе возникает конфликтная ситуация в случае обращения двух или более процессоров одновременно к одному модулю памяти. При такой ситуации очередность обслуживания определяется службой приоритетов арбитра. Целью статьи является определение, справляется ли заданная многопроцессорная система с потоком задач, а также определение минимального числа процессоров, необходимого для решения конкретного потока задач. В заключении статьи отмечено, что можно построить имитационную модель многопроцессорной вычислительной системы с общей многомодульной памятью, используя принятые допущения и полученные в работе выражения для определения времен состояний моделей элементов системы.

Ключевые слова. Многопроцессорная система, многомодульная память, конфликт, арбитр, модель, программа.

ANALYSIS OF MULTIPROCESSOR SYSTEMS WITH SHARED MULTI-MODULE MEMORY

A.I. Martyshkin

*Penza State Technological University,
Russia, Penza, Alexey314@yandex.ru*

Abstract. The article presents the structure of a multiprocessor system with shared multi-module memory. It is shown that in such a system there is a conflict situation in the case of two or more processors simultaneously to the same memory module. In this situation, the priority of service is determined by the arbitrator's priority service. The purpose of the article is to determine whether a given multiprocessor system copes with a task flow, as well as to determine the minimum number of processors required to solve a particular task flow. In conclusion of the article it is noted that it is possible to build a simulation model of a multiprocessor computing system with a common multi-module memory, using the assumed assumptions and the expressions obtained in the work to determine the state times of the models of the system elements.

Keywords. Multiprocessor system, multi-module memory, conflict, arbiter, model, program.

Объектом рассмотрения статьи предлагается многопроцессорная система (МПС) с общей многомодульной памятью (ОМП) (рисунок 1). В состав МПС входят N блоков процессоров (ЦП) (P) со своей кэш-памятью (CM), M блоков общей памяти (MM) и коммутаторов K . Количество модулей памяти (MP) не будет превышать число ЦП в системе ($M \leq N$).

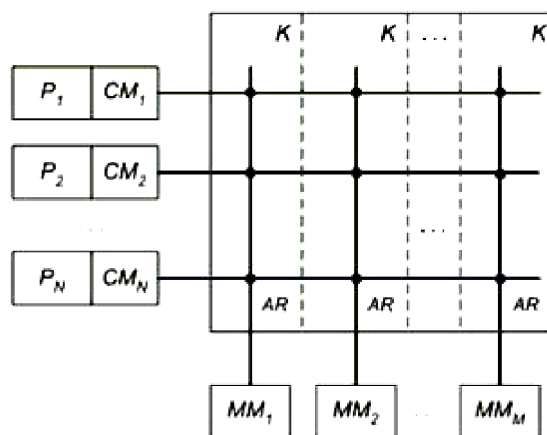


Рис. 1. Структура МПС с ОМП.

AR – арбитр; CM – кэш-память; K – коммутатор; MM – модуль общей памяти; P – процессор

В такой МПС возможно возникновение конфликтов в случае, когда два или более ЦП обращаются к одному МП одновременно [1]. При возникновении такой ситуации первоочередность обслуживания определяется выставлением приоритета арбитром (AR), входящего в состав коммутатора. Известны три различных приоритета: фиксированный, циклический и случайный [2, 3]. В самом общем случае, на месте некоторых ЦП включены устройства ввода/вывода (УВВ) с контроллерами прямого доступа к памяти (ПДП или КПДП). Они наряду с ЦП, являются активными элементами системы. УВВ с КПДП можно рассматривать как ЦП ввода-вывода. Каждый МП выполняется в виде секции, число которых определяется размером блока кэш-памяти. Способ формирования адресов МП подробно описан в [4].

Назначение рассматриваемой МПС – решение задач из потока, состоящего из различных типов данных. В статье принято, что команды обработки задач заранее записаны в MM . Данные для определенной задачи поступают через конкретное УВВ в фиксированные интервалы времени. Любая задача описывается графом, описанным при помощи ярусно-параллельной формы (ЯПФ), где каждый ее сегмент выполняется на одном из ЦП. Сегменты графа имеют возможность параллельного выполнения, если в системе есть свободные ЦП. Сегмент готов к выполнению, если готовы все данные для его решения. Для распределения на выполнение готовых сегментов задач по ЦП выделяется управляющий процессор (УЦП), остальные – рабочие процессоры (РЦП) – предназначенные для непосредственного решения сегментов задач.

Цель создания модели МПС – определить, справится ли конкретная МПС с поступающим входящим потоком задач, и кроме того, определить минимальный набор числа ЦП, который необходим для решения определенного (заданного) потока задач.

Модель любой системы, как правило, строится из набора моделей ее элементов и правил внутреннего взаимодействия. Выбор элементов определяется требуемой степенью детализации. Способы и правила взаимодействия моделей должны отражать реальные взаимодействия элементов системы на выбранном уровне детализации [5]. Модель элемента системы может находиться в определенном состоянии, которому соответствует некоторый процесс в конкретном элементе системы. Число возможных состояний модели определяется числом выполняемых действий соответствующих элементов системы. Модель вполне может переходить из состояния в состояние, что означает переход элемента системы от выполнения одного действия к другому. Смена состояний модели – событие. В отличие от реального перехода элемента системы от одного действия к другому, длительность события принимается равной нулю.

Любая конкретная модель элемента, как правило, характеризуется набором своих состояний, правилом смены состояний и строго уникальным идентификатором, определяющим модель элемента в модели системы [6]. Состояние модели всей системы в определенный мо-

мент времени характеризуется набором состояний моделей ее элементов в этот момент. Для возможности учета длительности состояний моделей элементов в модель системы вводится так называемый счетчик текущего времени (СТВ). Значение СТВ может выражаться в реальных единицах времени (секундах, минутах и т. д.), но также может выражаться и в модельных единицах времени, в этом случае должно быть установлено соответствие между реальными и модельными единицами. Событие, как правило, может сопровождаться передачей сообщений другим моделям элементов МПС. Эти сообщения должны содержать уникальные идентификаторы, характеризующие их типы и параметры.

Исследуемыми элементами системы в данном случае являются: РЦП с кэш-памятью, УП, УВВ и ММ с коммутатором и арбитром. ММ с коммутатором и арбитром – целостный элемент в модели, т.к. характеризуются единым набором состояний, определяемым арбитром. Главный (управляющий) ЦП принимает сигналы от УВВ, информирующие об окончании записи входных данных задачи и сигналы, информирующие об окончании решения сегментов от рабочих ЦП. Функции главного ЦП: выбор готового к выполнению сегмента, выбор свободного ЦП и назначение выбранного сегмента на свободный ЦП, которое заключается в отправке сигнала рабочему ЦП с указанием номера задачи и сегмента.

Рабочий ЦП решает закрепленный за ним сегмент задачи и по окончании решения сегмента задачи отправляет сигнал УП. Для описания алгоритма решения задачи РЦП приняты следующие допущения:

- РЦП считывает входные данные сегмента из ММ в свою кэш-память перед решением;
- запись полученных результатов в ОМП осуществляется строго после решения сегмента.

Время считывания программы определяется из некоторых предположений. Примем, что команда в среднем выполняется за 2 такта. Тогда количество выполняемых команд $N_{ВЫП}$ за время решения сегмента будет определено согласно выражению

$$N_{ВЫП} = \frac{T_{SEG}}{2 \cdot T_{CY}}, \quad (1)$$

где T_{SEG} – собственно время решения сегмента;

T_{CY} – длительность одного такта.

Число считываемых команд из ОМП меньше этого значения, т.к. одни и те же команды находясь в кэш-памяти могут выполняться несколько раз. Для вычисления числа считываемых из ОМП команд определим коэффициент цикличности $K_{ЦИКЛ}$ (например, 2). Тогда количество считываемых из ОМП команд $N_{СЧИТ}$ определится согласно выражению

$$N_{СЧИТ} = \frac{N_{ВЫП}}{K_{ЦИКЛ}} = \frac{T_{SEG}}{4 \cdot T_{CY}}. \quad (2)$$

Если принять допущение, что средняя длина команды равна одному слову, то значение $N_{СЧИТ}$ будет равно числу считываемых слов.

Команды и данные сегмента считываются блоками. Последовательно идущие блоки располагаются в различных МП, доступом к каждому управляет соответствующий арбитр, входящий в состав коммутатора. Поступление задач в МПС осуществляется через УВВ. Задача определенного типа приходит через закрепленное УВВ, т.е. распределение задач по УВВ происходит статически. Перед записью новых входных данных УВВ считывает резуль-

тат решения предыдущей задачи. Взаимодействие UBB с ОМП осуществляется по словам в режиме ПДП на частоте работы ОМП. Время T_{IN} , необходимое для записи новых входных данных задачи объемом V_{IN} слов, определяется согласно выражению

$$T_{IN} = V_{IN} \cdot T_{CY}. \quad (3)$$

Время T_{OUT} , затрачиваемое на получение результатов решения задачи объемом V_{OUT} слов, определяется согласно выражению

$$T_{OUT} = (V_{OUT} + 1) \cdot T_{CY}. \quad (4)$$

Резюмируя все вышеперечисленное следует отметить, что можно построить имитационную модель МПС с ОМП, используя допущения, принятые для РЦП и применяя выражения 1-4 для определения времен состояний моделей элементов МПС.

Библиографический список

1. Мартышкин А.И. Исследование подсистем памяти с буферизацией транзакций на моделях массового обслуживания // XXI век: итоги прошлого и проблемы настоящего плюс. – 2011. – № 3 (03). – С. 124-131.
2. Мартышкин А.И. Варианты реализации и структура аппаратного арбитра общей шины многопроцессорной системы // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 2 (22). – С. 195-204.
3. Мартышкин А.И. Функциональная организация аппаратного арбитра общей шины многопроцессорной системы // Модели, системы, сети в экономике, технике, природе и обществе. – 2017. – № 3 (23). – С. 151-159.
4. Гилев Ю.М., Дерюгин А.А. Особенности работы многопроцессорной вычислительной системы с общей многомодульной памятью /Международный форум информатизации – 2002: Доклады международной конференции «Информационные средства и технологии». 14-16 октября 2003 г., в 3-х т.т. Т3. – М.: Янус-К, 2003. – 221 с.
5. Марков А.А. Моделирование информационно-вычислительных процессов: Учебное пособие для вузов. – М.: Изд-во МГТУ им. Н.Э. Баумана, 1999. – 360 с.
6. Мартышкин А.И. К вопросу оценки времени обслуживания заявок при выполнении операций обмена в многопроцессорных системах на кристалле с разделяемой памятью // Приоритеты мировой науки: эксперимент и научная дискуссия Материалы X международной научной конференции. – 2016. – С. 81-87.

УДК 004.31; ГРНТИ 50.33.04

НАТУРНЫЙ ЭКСПЕРИМЕНТ ПО ИССЛЕДОВАНИЮ ОБРАЗЦА РЕКОНФИГУРИРУЕМОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛА НА БАЗЕ ПЛИС

А.И. Мартышкин

*Пензенский государственный технологический университет,
Россия, Пенза, Alexey314@yandex.ru*

Аннотация. В статье проводится экспериментальное исследование образца реконфигурируемой вычислительной системы для цифровой обработки сигнала на базе ПЛИС. В статье показана 4-х процессорная РВС, реализованных на FPGA. Для верификации разработанного алгоритма и полученных временных диаграмм в среде моделирования ModelSim-Altera 10.0с, проведен эксперимент на аппаратной платформе РВС. По результатам эксперимента можно принять, что алгоритм работы устройства работоспособен.

Ключевые слова. Реконфигурируемая система, диспетчер задач, аппаратная реализация, память, быстродействие, синхронизация, эксперимент, верификация.

A NATIONAL EXPERIMENT FOR INVESTIGATION OF A SAMPLE OF A RECONFIGURABLE COMPUTING SYSTEM FOR DIGITAL PROCESSING OF SIGNAL ON A FPGA BASIS SYSTEM

A.I. Martyshkin

Penza State Technological University,

Russia, Penza, Alexey314@yandex.ru

Abstract. The article presents an experimental study of a model of a reconfigurable computing system for digital signal processing based on FPGA. The article shows a 4-processor PBC implemented on FPGA. To verify the developed algorithm and the obtained time diagrams in The ModelSim-Altera 10.0 C simulation environment, an experiment was carried out on the RVS hardware platform. According to the results of the experiment, it can be accepted that the algorithm of the device is working.

Keywords. Reconfigurable system, task manager, hardware implementation, memory, performance, synchronization, experiment, verification.

Распараллеливание вычислительных процессов – одно из возможных решений проблемы повышения производительности. Однако производительность стандартных методов построения параллельных вычислений, обычно не превышает 10-15 % от указанной пиковой производительности. Это обусловлено тем, что жесткая архитектура ВС часто не соответствует информационной структуре решаемой задачи. Для преодоления сложившейся ситуации и используется способ построения гибкой реконфигурируемой системы на базе FPGA (Field-Programmable Gate Array) с управляющими процессорами [1, 3, 4].

Разработка и увеличение логики в FPGA позволяет применять более сложные алгоритмы, которые можно запрограммировать в микросхему. Подключение такого типа FPGA к современному процессору через высокоскоростную шину, например, PCI Express, позволило конфигурируемой логике действовать больше как сопроцессор, а не периферийное устройство. Это привело к реконфигурируемости в сфере высокопроизводительных вычислений.

В данной статье PBC это устройство, состоящее из 4 реконфигурируемых процессоров, реализованных на FPGA [2, 6]. Разработка подобной системы на FPGA предоставляет возможность для реконфигурации устройства под различные классы задач цифровой обработки сигнала, и обработки больших массивов данных.

Разрабатывая многопроцессорную вычислительную систему (МПВС), инженеры сталкиваются с проблемой, уменьшения потерь при планировании процессов. Назначение процессов или потоков по процессорным узлам в планировщике выполняет функция диспетчеризации задач [7]. В МПВС с глобальной очередью диспетчер выполняется программно посредством вызова этих функций из общей памяти, где расположена программа ОС.

Являясь достаточно быстрой, программная реализация в пространстве пользователя имеет сложную структуру, так как для произведения сопряженной с диспетчеризацией процедуры синхронизации необходимо 3 семафора. Первый счетчик, который считает число мест, занятых готовых к обработке процессами, второй счетчик, считает количество процессоров, находящихся на обслуживании, третий это мьютекс для функции взаимного исключения, останавливающего одновременный доступ некоторых свободных процессоров к одной очереди, которая является в виде общего ресурса.

Чтобы разрешить этот вопрос используют метод синхронизации в пространстве ядра, но повышенные временные затраты сильно уменьшают производительность МПВС.

Оптимальное решение вышеописанных проблем заключается в аппаратной реализации функции синхронизации, так как это снимает ответственность за выполнение данных функции с процессорных узлов, что повышает производительность, достигается увеличение надежности ОС. Предложенный метод основан на том, что функция диспетчера задач будет выполнена в виде независимого специализированного процессора в составе реконфигурируемой вычислительной системы.

Для экспериментального моделирования, спроектированного ДЗ, были созданы блоки имитации 4 процессоров и блок генератора задач. Моделирование проводилось с помощью ПО ModelSim-Altera 10.0с. [5, 8] После компиляции была запущена симуляция проекта. Также проводилось моделирование с помощью лабораторного стенда, который включает в себя логический анализатор АКИП-9101, опытный образец разрабатываемой РВС (рисунок 2). По результатам проведения экспериментов были получены временные диаграммы (рисунки 1,3) работы системы с диспетчером задач.

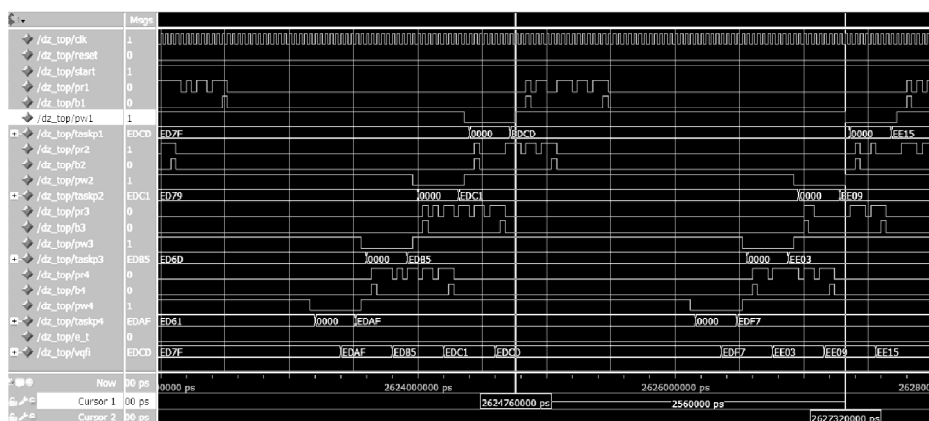


Рис. 1. Диаграммы работы системы в среде моделирования ModelSim-Altera 10.0с

Представленные временные диаграммы на рисунке 1 показывают, что идентификатор поступившей задачи под номером EDAF передан на обработку в процессор под номером 4 (taskp4). Процессор под номером 4 оказался занятым, поэтому следующий идентификатор задачи EDB5 принят на обработку в третий процессор.

По результатам моделирования видно, что первым для обслуживания текущей задачи назначен четвертый процессор, следом назначается третий процессор, и так далее. Это обусловлено схемой приоритетов от выполнения задач в системе, т.е. от четвертого процессора к первому. Для верификации разработанного алгоритма и полученных временных диаграмм в среде моделирования ModelSim-Altera 10.0с, проведен эксперимент на основе аппаратной платформы опытного образца РВС.



Рис. 2. Стенд для проведения эксперимента

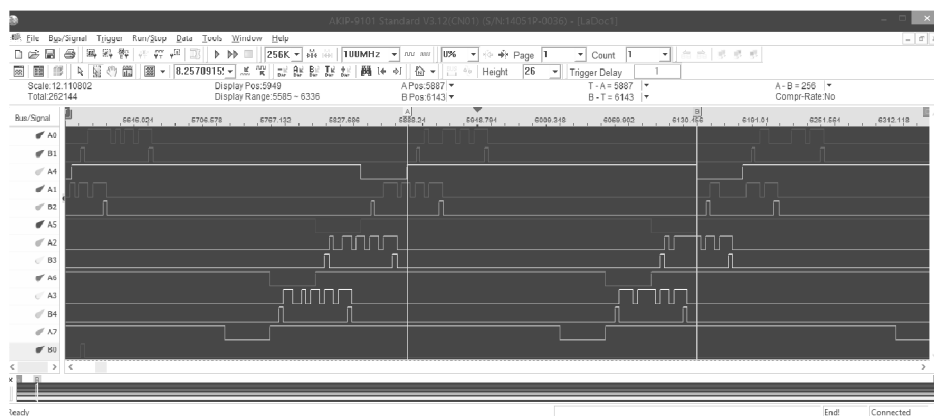


Рис. 3. Временные диаграммы, полученные с помощью логического анализатора

По результатам эксперимента можно заключить, что алгоритм работы диспетчера действительно работает, что видно из временных показаний выполнения работы процессора (сигнал rw1, сигнал A4 на рисунках 1 и 3 соответственно) который составляет 256 тактовых отсчетов или 2560 нс.

Библиографический список

1. Мартенс-Атюшев Д.С., Мартышкин А.И. Разработка и исследование модуля подсистемы диспетчеризации задач реконфигурируемой вычислительной системы для цифровой обработки сигнала // Международный студенческий научный вестник. – 2017. – № 4-9. – С. 1411-1414.
2. Мартенс-Атюшев Д.С., Мартышкин А.И. Разработка и исследование реконфигурируемого вычислительного кластера для цифровой обработки сигнала // Современные информационные технологии. – 2015. – № 21. – С. 190-195.
3. Мартенс-Атюшев Д.С., Мартышкин А.И. Разработка и исследование реконфигурируемой системы для цифровой обработки сигнала // Международный студенческий научный вестник. – 2016. – № 3-1. – С. 86-88.
4. Мартенс-Атюшев Д.С., Мартышкин А.И. Реконфигурируемый вычислительный кластер для цифровой обработки сигнала // Современные методы и средства обработки пространственно-временных сигналов: Сборник статей XIII Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2015. – С. 112-117.
5. Мартенс-Атюшев Д.С., Мартышкин А.И. Эксперимент по исследованию подсистемы планирования реконфигурируемой вычислительной системы для цифровой обработки сигнала // Международный студенческий научный вестник. – 2017. – № 4-9. – С. 1408-1410.
6. Мартышкин А.И., Мартенс-Атюшев Д.С., Маркин Е.И. К вопросу построения реконфигурируемой вычислительной системы на базе ПЛИС для цифровой обработки сигнала // Современные инновационные технологии подготовки инженерных кадров для горной промышленности и транспорта. – 2017. – № 4. – С. 433-439.
7. Мартышкин А.И., Мартенс-Атюшев Д.С. Разработка подсистемы планирования и назначения задач реконфигурируемой вычислительной системы для цифровой обработки сигнала // Современные методы и средства обработки пространственно-временных сигналов: Сборник статей XIV Всероссийской научно-технической конференции. Под редакцией И.И. Сальникова. – 2016. – С. 115-119.
8. Мартышкин А.И. Реализация опытного образца реконфигурируемой вычислительной системы для цифровой обработки сигнала на базе программируемых логических интегральных схем // Новые информационные технологии и системы: Сборник научных статей XIV Международной научно-технической конференции, посвященной 70-летию кафедры «Вычислительная техника» и 30-летию кафедры «Системы автоматизированного проектирования». Пенза, – 2017. – С. 243-246.

УДК 004.94; ГРНТИ 50.05.19

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ С ПРИМЕНЕНИЕМ ФУНКЦИОНАЛЬНЫХ БЛОКОВ

А.И. Мартышкин, Д.С. Мартенс-Атюшев

Пензенский государственный технологический университет,

Россия, Пенза, Alexey314@yandex.ru

Аннотация. В статье описывается вариант создания имитационной модели распределенной системы управления с использованием функциональных блоков. Показан пример разработки системы массового обслуживания с применением абстрактных функциональных блоков и языка XML. Подробно описан блок обслуживающего прибора и блок типа «очередь». Все блоки показаны в общем виде. В конце работы приводятся основные результаты.

Ключевые слова. Система массового обслуживания, функциональный блок, имитационное моделирование, язык XML, обслуживающий прибор.

SIMULATION OF A DISTRIBUTED SYSTEM CONTROL USING FUNCTIONAL UNITS

A.I. Martyshkin, D.S. Martens-Atushev

Penza State Technological University,

Russia, Penza, Alexey314@yandex.ru

Abstract. The article describes the option of creating a simulation model of a distributed control system using functional blocks. Shows an example of development of system of mass service with the use of abstract functional blocks and XML language. Describes in detail the maintenance unit of the appliance and unit type "queue". All the blocks shown in the General view. The main results are presented at the end of the work.

Keywords. Queueing system, functional unit, simulation modeling, language XML, the device operator.

Развивающимся перспективным направлением в проектировании систем управления является методика проектирования на основе абстрактных функциональных блоков (ФБ) [1]. Каждый блок отвечает за выполнение какой-либо части управляющего алгоритма. Существует возможность создания своих блоков их последующего использования при проектировании, а также расширения функциональности уже существующих [2]. ФБ представляет собой программный модуль, выполняющий определенный алгоритм при поступлении события на соответствующий вход этого блока. Для описания типов ФБ используются языковые средства, описанные в стандарте IEC 61131, а также язык XML. Взаимодействие функциональных блоков основано на передаче управляющих сигналов – событий через соединение событий и передаче необходимых данных через соединение данных. ФБ описывается объявлением внешнего интерфейса, диаграммой управления исполнением и алгоритмами, выполняемыми в соответствии с диаграммой. На сегодняшний день существует несколько программных продуктов поддерживающих стандарт IEC61499 среди них можно выделить среду Function Block Development Kit (FBDK) и программный продукт IsaGRAF 5 [3]. Использование средств имитационного моделирования при проектировании распределенных систем управления с применением методики на основе абстрактных функциональных блоков дает возможность получать информацию о поведении, как отдельного элемента системы, так и о системе в целом, что позволяет упростить и сократить цикл разработки распределенных систем управления.

Рассмотрим конкретный пример разработки с использованием ФБ системы массового обслуживания, состоящей из очереди и обслуживающего прибора. Блок очереди (QUEUE) представлен на рисунке 1.

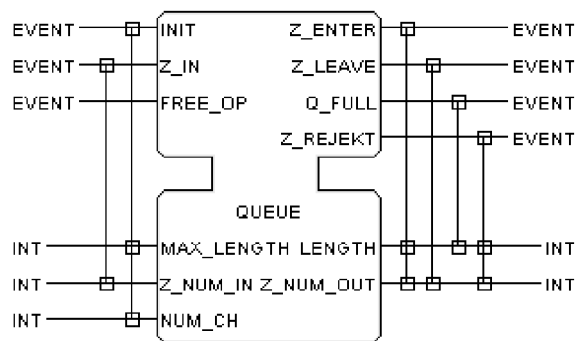


Рис. 1. Функциональный блок типа “ОЧЕРЕДЬ”

Он предназначен для сбора статистических данных и управления поступлением заявок извне, а также управляет загрузкой заявки из очереди в обслуживающий прибор. Для включения новой заявки в очередь используется событийный вход Z_IN, управляющий входом поступления заявок Z_NUM_IN. Этим же сигналом устанавливается количество каналов соответствующего обслуживающего прибора равное значению на входе NUM_CH. Событийный вход INIT используется для задания длины очереди MAX_LENGTH.

Событие на выходе Z_ENTER, предназначенном для уведомления блока статистики о появлении в очереди новой заявки. Выход данных LENGTH используется для выдачи, текущей длины очереди, а на выходе Z_NUM_OUT – значение номера заявки присвоенного ей на входе Z_NUM_IN. Событие на выходе Z_LEAVE появляется, когда обслуживающий прибор пуст и готов к обслуживанию новой заявки. Событие FREE_OP сообщает об освобождении обслуживающего прибора. Событие на выходе Q_FULL уведомляет блок статистики о достижении очередью состояния, когда очередь занята полностью. Выход события Z_REJEKT сигнализирует о том, что вновь поступившая заявка застигла очередь заполненной и была удалена из СМО без обслуживания.

Программа, описывающая функционирование ФБ QUEUE, представлена на языке XML.

```
<FBType Name="QUEUE" Comment="Model for a Rotating Shaft" >
  <InterfaceList>
    <EventInputs>
      <Event Name="INIT" Comment="Запуск инициализации, установка максимальной длины очереди" >
        <With Var="NUM_CH" />
        <With Var="MAX_LENGTH" />
      </Event>
      <Event Name="Z_IN" Comment="Появление заявки" >
        <With Var="Z_NUM_IN" />
      </Event>
      <Event Name="FREE_OP" Comment="Обслуживающий прибор свободен" >
      </Event>
    </EventInputs>
    <EventOutputs>
      <Event Name="Z_ENTER" Comment="заявка вошла в очередь" >
        <With Var="LENGTH" />
        <With Var="Z_NUM_OUT" />
      </Event>
      <Event Name="Z_LEAVE" Comment="Заявка покинула очередь" >
        <With Var="Z_NUM_OUT" />
      </Event>
      <Event Name="Q_FULL" Comment="Очередь заполнена" >
        <With Var="LENGTH" />
      </Event>
      <Event Name="Z_REJEKT" Comment="Заявка отклонена" >
      </Event>
    </EventOutputs>
  </InterfaceList>
</FBType>
```

```

    <With Var="LENGTH" />
    <With Var="Z_NUM_OUT" />
  </Event>
</EventOutputs>
<InputVars>
  <VarDeclaration Name="MAX_LENGTH" Type="INT" Comment="Максимальная длина очереди" />
  <VarDeclaration Name="Z_NUM_IN" Type="INT" Comment="Номер поступающей заявки" />
  <VarDeclaration Name="NUM_CH" Type="INT" Comment="Количество каналов в соответствующем
обслуживающем приборе" />
</InputVars>
<OutputVars>
  <VarDeclaration Name="LENGTH" Type="INT" Comment="Текущая длина очереди"/>
  <VarDeclaration Name="Z_NUM_OUT" Type="INT" Comment="Номер заявки покидающей очередь" />
</OutputVars>
</InterfaceList>
</FBType>

```

Функциональный блок обслуживающего прибора представлен на рисунке 2.

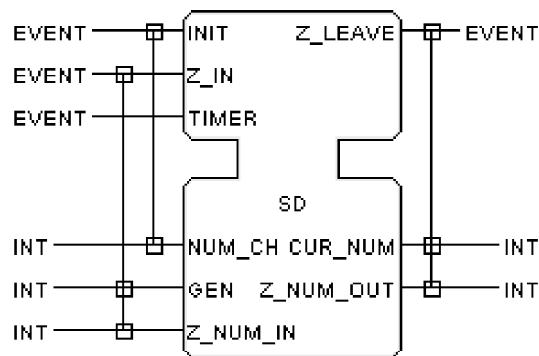


Рис. 2. Функциональный блок обслуживающего прибора

Вход данных NUM_CH используется для задания количества каналов прибора. Для включения заявки в обслуживающий прибор используется вход Z_IN. При появлении здесь события считывается значение на входе данных Z_NUM_IN, содержащее номер заявки, а также устанавливается время обработки заявки. Вход TIMER используется для реализации задержки подсчетом событий от таймера. После завершения обработки заявки выдается событие на выходе Z_LEAVE и выдается номер заявки на выходе Z_NUM_OUT, а также выдается текущее значение количества занятых каналов с выхода CUR_NUM.

Программа, описывающая функционирование ФБ SD, представлена на языке XML.

```

<FBType Name="SD" Comment="Блока обслуживающего прибора" >
  <InterfaceList>
    <EventInputs>
      <Event Name="INIT" Comment="Запуск инициализации, установка максимальной длины очереди" >
        <With Var="NUM_CH" />
      </Event>
      <Event Name="Z_IN" Comment="Появление заявки" >
        <With Var="GEN" />
        <With Var="Z_NUM_IN" />
      </Event>
      <Event Name="TIMER" Comment="Обслуживающий прибор свободен" >
      </Event>
    </EventInputs>
    <EventOutputs>
      <Event Name="Z_LEAVE" Comment="заявка закончила обслуживание" >

```

```

    <With Var="Z_NUM_OUT" />
    <With Var="CUR_NUM" />
  </Event>
</EventOutputs>
<InputVars>
  <VarDeclaration Name="NUM_CH" Type="INT" Comment="Количество каналов" />
  <VarDeclaration Name="GEN" Type="INT" Comment="Количество каналов"/>
  <VarDeclaration Name="Z_NUM_IN" Type="INT" Comment="Имя заявки поступившей на
обслуживание" />
</InputVars>
<OutputVars>
  <VarDeclaration Name="CUR_NUM" Type="INT" Comment="Текущее количество занятых каналов" />
  <VarDeclaration Name="Z_NUM_OUT" Type="INT" Comment="Имя выходящей заявки" />
</OutputVars>
</InterfaceList>
</FBType>

```

Библиографический список

1. Елькин И.В., Кустарев П.В. Модель абстрактных функциональных блоков // Научно-технический вестник СПбГИТМО (ТУ). Выпуск 10. Информация и управление в технических системах. - СПб.: СПбГИТМО(ТУ), - 2003. - С. 55-61.
2. Мартышкин А.И. Имитационная модель распределенной системы с применением абстрактных функциональных блоков // Информационные технологии в экономических и технических задачах: Сборник научных трудов Международной научно-практической конференции. - 2016. - С. 261-263.
3. Мартышкин А.И. Создание имитационной математической модели распределенной вычислительной системы с использованием абстрактных функциональных блоков // Новая наука: история становления, современное состояние, перспективы развития: Сборник статей Международной научно-практической конференции: в 2-х частях. - 2016. - С. 41-43.

УДК 004.021; ГРНТИ 50.41.25

МОДИФИКАЦИЯ СТЕГАНОГРАФИЧЕСКОГО МЕТОДА LSB ДЛЯ ПОВЫШЕНИЯ СЕКРЕТНОСТИ ПЕРЕДАЧИ СООБЩЕНИЯ

А.В. Гиголаев, Н.А. Тярт, О.Г. Швечкова

*Рязанский государственный радиотехнический университет,
Россия, Рязань, tyarnikita@gmail.com*

Аннотация. Рассматриваются вопросы шифрования и сокрытия информации, методы стеганографии. Предлагается модификация метода LSB с целью повышения эффективности сокрытия информации. Показывается возможность использования протокола ECDH в сочетании с выбранным методом. Анализируются достоинства и недостатки предложенного решения.

Ключевые слова. Стеганография, сокрытие информации, LSB, ECDH, обработка изображений

MODIFICATION OF STEGANOGRAPHIC LSB METHOD TO ENHANCE THE SECRECY OF THE MESSAGE TRANSMISSION

A.V. Gigolaev, N.A. Tyart, O.G. Shvechkova

*Ryazan State Radio Engineering University,
Ryazan, Russia, tyarnikita@gmail.com*

Abstract. The issues of encryption and concealment of information, methods of steganography are considered. A modification of the LSB method with the aim of improving the effectiveness of information hiding is proposed. The possibility of using the ECDH protocol in combination with the chosen method is shown. The advantages and disadvantages of the proposed solution are analyzed.

Keywords. Steganography, information hiding, LSB, ECDH, image processing

В настоящее время сфера информатизации служит фундаментальным фактором развития всех отраслей человеческой деятельности. Одной из самых важных областей данной сферы является организация передачи, хранения и обработки данных. Информация, циркулирующая на различных этапах информационного обмена, не всегда является общедоступной, обладает различными уровнями конфиденциальности и ценности для пользователя. Как следствие, возникает проблема защиты данных от несанкционированного доступа. При этом методы и средства обеспечения безопасности информации и информационных систем разнятся и, как правило, ориентированы на отдельные классы решаемых задач и соответствующие предметные области. Общепринят тезис о невозможности существования абсолютной защиты, поэтому механизмы безопасности выбираются в соответствии с запросами пользователя, ценностью, сроком жизни информации и ориентированности на конкретную область практического применения. Ключевым фактором при выборе методов и средств обеспечения безопасности информации является анализ угроз и рисков в конкретной информационной системе. Исходя из совокупности перечисленных факторов, выбирается тот механизм защиты, который будет учитывать приведенный перечень исходных параметров, предпосылок, особенностей объектов защиты и будет ориентирован на защиту от какого-либо конкретного вида угроз, т.е. конфиденциальности, целостности, доступности [1].

Надежность методов разграничения доступа к конфиденциальной информации и шифрования данных определяются сложностью математического аппарата, лежащего в основе алгоритмов, и качеством разработки их программной реализации, стоимостью внедрения, эксплуатации и поддержки. Как известно из практики, методы проверки прав доступа возможно обойти, а даже самый сложный шифр – взломать. Поэтому, даже с учетом значительных финансовых возможностей, надежных алгоритмов и искусного исполнения программного кода, конкретные задачи пользователя по защите информации не всегда удается успешно решить. Из необходимости решения таких специфических задач возникло одно из направлений защиты информации, разновидность методов криптографии – стеганография. Это научное направление позволяет обеспечить не только тайную передачу данных, но и скрывает сам факт такой передачи. При этом вероятность взлома злоумышленником значительно снижается за счет сокрытия самого факта наличия секретной скрытой информации.

Для организации стеганографического канала тайной передачи информации необходимы два условия: наличие исходного секретного сообщения, в качестве которого может выступать текст, изображение или аудиоданные, подлежащего сокрытию, и контейнера – любой информации, предназначенной для помещения внутрь нее запрятываемого сообщения.

Методы стеганографии ориентированы на решение специфического круга задач, одной из которых является сокрытие самого факта передачи информации, при этом контейнером для запрятывания информации может служить обычный файл растрового изображения в любом из популярных форматов (например, JPEG, PNG или BMP). Другое популярное применение методов стеганографии – внедрение цифровой подписи в контейнер с целью защиты авторских прав.

Все методы сокрытия информации в контейнерах-изображениях можно разделить на три типа:

1. Цифровые методы, работающие с двоичным представлением изображения в памяти ЭВМ. Обладают хорошей точностью и детерминированностью алгоритмов, а также приводят к минимальным изменениям контейнера.
2. Частотные методы, влияющие на цветовые характеристики изображения. Например, нужным образом изменяется яркость выбранных областей изображения. Обладают хорошей устойчивостью к непредвиденным искажениям контейнера (сжатию, помехам, аффинным преобразованиям), но влияют на контейнер значительно сильнее.

3. Файловые методы, использующие известные особенности форматов файлов. Например, информация может быть внесена в метаданные контейнера. Не обеспечивают хорошей скрытности, могут сильно увеличивать вес контейнера.

Решению основной задачи стеганографии – сокрытию наличия тайного сообщения в контейнере – в большей степени соответствуют методы первого типа. При их использовании видимые изменения контейнера минимальны, следовательно – меньше вероятность обнаружения скрытого сообщения.

Одним из наиболее известных методов стеганографии является метод LSB (Least Significant Bit, Наименьший Значащий Бит) [2]. Суть метода запрятывания состоит в размещении битов скрываемого сообщения в младшие биты каждого пикселя контейнера. Особенностью алгоритма является использование только синего канала пикселя, так как глаз человека наименее чувствителен к искажениям именно в этом канале. Достоинством метода следует считать возможность свести к минимуму искажения контейнера [3]. Несмотря на отмеченные преимущества данного метода, алгоритм LSB имеет ряд недостатков. К таковым можно отнести:

1. Отсутствие какой-либо устойчивости к помехам при передаче контейнера, к сжатию изображения.

2. Легкое обнаружение факта вложения сообщения в контейнер при помощи статистического анализа контейнера на ЭВМ в автоматическом режиме (для обнаружения достаточно простой программы-фильтра).

3. Отсутствие шифрования сообщения.

Исходя из анализа недостатков рассматриваемого метода, с целью устранения выявленных проблем и повышения эффективности алгоритма, авторами предлагается усовершенствование алгоритма LSB путем модификации процедуры вложения сообщения в контейнер и дополнительной процедуры шифрования сообщения.

Предлагаемое решение задачи повышения эффективности рассматриваемого алгоритма предполагается получить за счет усиления двух факторов:

1. Результат сокрытия информации в стего-контейнере незаметен при визуальном восприятии человеком и малоопределяем при машинном анализе.

2. Повышается сложность процедуры корректного изъятия сообщения из контейнера без знания алгоритма.

Одним из важных этапов предлагаемого модифицированного алгоритма является получение секретного ключа. Для этого выбран протокол ECDH (Elliptic curve Diffie–Hellman, Протокол Диффи-Хеллмана на эллиптических кривых) [4]. Сгенерированный секретный ключ может быть использован как для шифрования сообщения, так и в алгоритмах сокрытия сообщения в контейнере.

Рассмотрим этапы реализации предлагаемого алгоритма.

1. До начала организации стегоканала участники информационного обмена договариваются о следующих параметрах:

- Максимальный размер сообщения p (в битах).
- Терминальный символ T , означающий конец сообщения.
- Параметры эллиптической кривой a, b .
- Точка G на эллиптической кривой – генератор ключа.
- Порядок подгруппы n .
- Параметр контейнера, выбранный в качестве открытого ключа (дата отправки сообщения, размер изображения в пикселях, др.).

2. Генерация секретного ключа S по протоколу ECDH.

3. Сжатие / шифрование скрываемой информации. При этом используется секретный ключ S , полученный на предыдущем шаге. Результат шифрования должен быть не больше оговоренного максимального размера p .

4. Выбор изображения-контейнера с количеством пикселей не меньше p . Желательно, чтобы размер контейнера был значительно больше размера скрываемого сообщения (в 10 раз и более).

5. Разбиение изображения на блоки. Количество блоков равно длине сообщения p . Размеры блоков равны (при неудобном размере изображения могут остаться узкие полосы незадействованных пикселей справа и снизу). Форма блоков стремится к квадратной. В одном блоке будет N пикселей.

6. Определение порядка обхода блоков. Это может быть сделано с помощью функции выбора $f(S, A_i) = i$, где S – секретный ключ, A_i – массив номеров неиспользованных до i -го шага блоков, i – номер выбранного блока. Соответственно, массив A_{i+1} будет состоять из всех элементов массива A_i , исключая использованный элемент i : $A_{i+1} = A_i \setminus i$.

7. Выбор для каждого блока числа используемых пикселей N_u : $1 \leq N_u \leq N$. Обозначим данный выбор функцией $g(S, N, i) = N_u$.

8. Определение порядка обхода пикселей. Операция аналогична определению выбора обхода блоков и описывается той же функцией $f(S, A_i) = i$, только в данном случае A_i – массив номеров пикселей, а не блоков. Другое различие – выбор происходит не до тех пор, пока массив A_i не опустеет, а останавливается после выбора необходимого количества N_u пикселей.

9. Выбор для каждого пикселя используемого в кодировании канала цвета (*Red*, *Green* или *Blue*; далее R , G или B). Канал прозрачности не используется, изменения в нем будут слишком заметны при машинном анализе. Для абсолютно черных и белых пикселей используется только синий канал (B) как наименее заметный. Обозначим данный выбор функцией $h(S, x, y) = c$, где x и y – координаты пикселя в изображении, $c \in \{R, G, B\}$ – выбранный канал цвета.

10. Вложение в каждый блок (в выбранном в п.5 порядке) одного бита скрываемого сообщения. То есть в i -й по очереди блок вкладывается i -й в сообщении бит. При этом данный бит заносится во все используемые пиксели данного блока в младший разряд выбранного для конкретного пикселя канала цвета.

Для расшифровывания сообщения на принимающей стороне повторяются пункты 2-9 для определения блоков, их порядка, пикселей и каналов цвета, содержащих информацию. Затем в рамках каждого блока определяется наиболее встречающийся закодированный бит (на случай помех), он и считается верным. Из всех бит, таким образом, складывается сообщение.

Сообщение может быть более короткой длины, чем оговоренная максимальная. С целью выравнивания длины сообщения до определенного максимального размера в конце помещается специально оговоренный терминальный символ, а недостающие позиции заполняются случайным шумом.

Резюмируя вышеизложенное, к результатам проделанной работы следует отнести.

Достоинства разработанной версии модифицированного стеганографического алгоритма LSB:

1. Высокая сложность функционирования алгоритма в совокупности, в частности из-за использования ECDH, что повышает надежность алгоритма, усложняет процесс взлома результатов кодирования.

2. Простота реализации алгоритма как совокупности несложных шагов.

3. Базовая устойчивость к сжатию изображения и помехам за счет избыточного блочного кодирования.

4. Сохранение достаточной эффективности в плане объема передаваемого сообщения. К примеру, при наличии изображения 500*500 пикселей и использовании блоков 4*4 пикселя максимальная длина сообщения составляет 15625 бит, или около 2 килобайт. Это приблизительно одна книжная страница текста (2000 символов в кодировке ANSI). При использовании дополнительного сжатия сообщения перед помещением в контейнер эффективность алгоритма возрастает (а также возрастет и устойчивость к обнаружению статистическими методами).

5. Минимально необходимый уровень изменения контейнера. При этом часть пикселей не выбирается для изменения вообще, а выбранные – изменяются только в 50% случаев.

Проведенный анализ полученных результатов выявил ряд нерешенных проблем разработанного алгоритма. К ним можно отнести:

1. Слабую устойчивость к сжатию изображения по сравнению с алгоритмами, основанными на частотных преобразованиях (например, дискретно-косинусным преобразованием).

2. Отсутствие устойчивости к любым аффинным преобразованиям (изменение размеров, поворот, отражение и т.п.).

3. Простое повторение бита в выбранных пикселях внутри блока может быть обнаружено с помощью статистического анализа.

Проведенные исследования на данном этапе завершаются доработкой программной реализации представленного алгоритма. После завершения этапа тестирования алгоритма на произвольных наборах данных будут получены количественные оценки повышения эффективности выполненной разработки по сравнению с базовой версией LSB, а также подтверждена его жизнеспособность в целом. Для устранения выявленных недостатков предполагается дальнейшая доработка алгоритма.

Практическая значимость алгоритма имеет следующие сферы применения:

1. Скрытая передача сообщений. Метод призван повысить скрытность факта наличия сообщения в контейнере. Скрытая передача открытого ключа в виде одного из параметров контейнера, генерация ключа с использованием аппарата эллиптических кривых, использование различных мест вложения битов, разных каналов цвета, зависимость алгоритмов от многих параметров – все это способствует достижению указанной цели.

2. Скрытая подпись изображений (защита авторских прав). Алгоритм имеет малую стойкость к изменениям контейнера, но зато практически не влияет на качество изображения. При этом требуется выбрать подходящую характеристику файла для использования в качестве открытого ключа, например – ширину и высоту изображения в пикселях.

Библиографический список

1. Евдокимова Л.М. Электронный документооборот и обеспечение безопасности стандартными средствами Windows: учеб. пособие / Л.М. Евдокимова, В.В. Корябкин, А.Н. Пылькин, О.Г. Швечкова. // М.: КУРС, 2017.
2. Коханович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. // К.: МК-Пресс, 2006.
3. Вовк О.О. Исследование стойкости методов скрытия информации в неподвижных изображениях. / О.О. Вовк, А.А. Астраханцев, А.В. Дорожан // Радиоэлектронные и компьютерные системы 2012, № 2 (54).
4. Мао, Венбо. Современная криптография: теория и практика: пер. с англ. // М.: Издательский дом «Вильямс», 2005.

УДК 004.42; ГРНТИ 50.41.29

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ «USER DEFINED TARGETS», «USER DEFINED OBJECTS» И «MULTI-TARGETS» В ПРИЛОЖЕНИЯХ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

В. Решетова, Е. Софьин

Рязанский государственный радиотехнический университет,

Рязань, arev62rus@mail.ru, eugenesofofin@gmail.com

Аннотация. В данной статье рассматривается объединение технологий «User Defined Targets», «User Defined Objects» и «Multi-Targets» для разработки приложения с поддержкой дополненной реальности.

Ключевые слова. AR приложение, технология UDT, технология UDO, технология Multi-Targets

USING «USER DEFINED TARGETS», «USER DEFINED OBJECTS» AND «MULTI-TARGETS» TECHNOLOGIES IN AR DEVELOPMENT

V. Reshetova, E. Sofin

Ryazan State Radioengineering University,

Ryazan, arev62rus@mail.ru, eugenesofofin@gmail.com

Abstract. This paper deals with the «User Defined Targets», «User Defined Objects» and «Multi-Targets» technologies in AR development.

Keywords. AR development, UDT technology, UDO technology, Multi-Targets technology

Введение

Дополненная реальность (augmented reality) – воспринимаемая смешанная реальность, создаваемая с использованием дополнительной информации [1]. Практически на все окружающие нас объекты могут накладываться изображения, анимация, 3D модели, текст и другая цифровая информация. Дополненная реальность формируется с помощью камеры смартфона, веб-камеры компьютера или другого устройства, которое может обрабатывать видео и аудио сигнал. Дополнительные данные всегда отображаются на изображении, называемом изображением-маркером.

Алгоритм распознавания маркеров

В настоящее время существует большое количество инструментов, которые предназначены для разработки приложений дополненной реальности. Наиболее известными являются следующие библиотеки: Vuforia, ArtToolKit, Wikitude, LayAR. Все они используют схожий алгоритм распознавания маркеров, который выглядит следующим образом.

1. Изображение с камеры переводится в цветовой режим «градации серого». Чаще всего используется следующее преобразование:

$$GS = (\max(R, G, B) + \min(R, G, B)) / 2 ;$$

2. Производится бинаризация изображения. Существуют различные подходы к бинаризации, которые условно можно разбить на две группы: пороговые и адаптивные. В случае с изображениями чаще всего применяют пороговую бинаризацию. При этом все пиксели изображения разбиваются на две группы (темные и светлые), в соответствии с устанавливаемым порогом яркости каждого пикселя. Пороговое изображение получается из исходного по следующему соотношению:

$$r(x, y) = \begin{cases} 1, S(x, y) > L \\ 0, S(x, y) \leq L \end{cases},$$

где $r(x, y)$ и $S(x, y)$ – уровни яркости пикселей изображения;
 L – пороговый уровень яркости.

3. На полученном изображении выделяются все замкнутые контуры. Как правило, при этом используется алгоритм заливки белых областей:

- 1) На исходном изображении выбирается случайный пиксель;
- 2) Координаты выбранного пикселя записываются в переменные x и y ;
- 3) Заполняется максимально возможный интервал, в котором находится выбранный пиксель. Заполнение происходит вправо и влево вплоть до граничных точек;
- 4) Крайняя левая и крайняя правая точка записываются в переменные x_l и x_r ;
- 5) Все вышеперечисленные действия повторяются над всеми строками пикселей, которые располагаются над и под интервалом (x_l, x_r) ;
- 6) Действия повторяются до тех пор, пока на изображении присутствуют незаполненные интервалы.

4. После преобразования на изображении выделяется самый большой замкнутый контур, который и будет являться изображением-маркером. Для выделения контуров применяют следующие алгоритмы: Marr-Hildreth Edge Detector, Canny Edge Detector, Boolean function based Edge Detector, Euclidian distance and Vector Angle based Edge Detector, Depth Edge Detection using Multi-Flash Imaging.

Наиболее же часто используют алгоритм Canny Edge Detector описываемый далее:

- 1) Сглаживание изображения. Происходит размытие изображения для устранения шума;
- 2) Поиск градиента. Границы отмечаются там, где градиент изображения приобретает максимальное значение;
- 3) Подавление значений, близких к максимумам. Только локальные максимумы отмечаются как границы;
- 4) Двойная пороговая фильтрация. Потенциальные границы определяются порогами;
- 5) Трассировка области неоднозначности. Итоговые границы определяются путем подавления всех краев, несвязанных с определенными границами.

Для того чтобы выделить определенный контур среди большого количества объектов, применяется алгоритм Рамера-Дугласа-Пекера, который позволяет уменьшить число точек кривой, аппроксимированной большей серией точек. Именно этот алгоритм делает возможным распознавание изображения-метки среди большого количества однотипных изображений.

Суть алгоритма состоит в том, чтобы по данной ломаной, аппроксимирующей кривую, построить ломаную с меньшим числом точек. Алгоритм определяет расхождение, которое вычисляется по максимальному расстоянию между исходной и упрощенной кривыми. Упрощенная кривая состоит из подмножества точек, которые определяются из исходной кривой.

5. Производится определение координат изображения-маркера и камеры используемого смартфона/компьютера. Схематично это можно представить так:

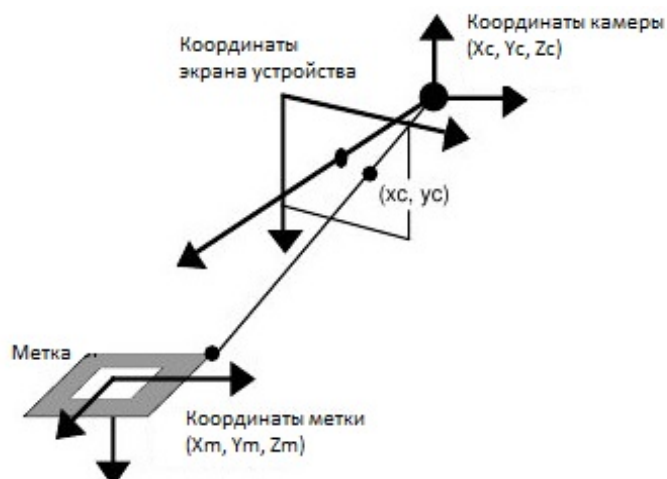


Рис. 1. Определение координат

Основная идея заключается в том, что при изменении угла обзора камеры используемого устройства происходит изменение размера и угла проекции [2].

Технология «User Defined Targets»

Большинство существующих AR инструментов позволяют распознавать только заранее определенные изображения-метки, которые задаются в процессе разработки приложения. Технология «User Defined Targets» (UTD) дает возможность создавать метки, используя камеру смартфона [3]. На сегодняшний день данная технология не получила широкого распространения и поддерживается только библиотекой Vuforia. Использование технологии UTD позволяет значительно расширить функционал разрабатываемого приложения.

Для наилучшего результата распознавания изображение-метка должно соответствовать следующим требованиям:

1. Метка должна иметь большое количество четко выраженных деталей;
2. Отчетливый контраст между светлыми и темными частями изображения;
3. Отсутствие повторяющихся деталей и узоров;
4. Небольшая дистанция до изображения (до 5 метров).

Технология «User Defined Objects»

Как говорилось ранее, технология дополненной реальности делает возможным наложение дополнительной информации (изображений, 3D моделей и пр.) на окружающие пользователя объекты. Чаще всего в AR приложениях применяется заранее определенный список объектов, которые пользователь может использовать для отображения поверх изображения-метки. Данный факт накладывает определенные ограничения. Например, при необходимости пользователь не сможет использовать изображения галереи своего устройства в качестве AR объекта.

Для решения этой проблемы применяется технология «User Defined Objects» (UDO), которая позволяет использовать пользовательские файлы в качестве объектов, накладываемых на изображения-метки. Главным минусом UDO является сложность реализации, так как способ доступа к файлам пользователя будет зависеть от типа устройства, его операционной системы и других факторов.

Технология «Multi-Targets»

Технология «Multi-Targets» предназначена для анализа и распознавания сразу нескольких изображений объектов. Это достигается за счет анализа всех замкнутых областей, которые попадают в объектив камеры. Также «Multi-Targets» можно использовать для распознавания объемных объектов. Например, для того чтобы использовать куб в качестве изображения-метки, достаточно сохранить изображение с каждой грани куба. После этого куб будет рассматриваться как единая метка. На практике же для более быстрого распознавания объемных объектов используется технология «Cubo Targets», которая является упрощенной версией «Multi-Targets» [4].

Для успешного распознавания нескольких изображений-меток они должны:

1. Иметь большое количество четко выраженных деталей;
2. Находиться в фокусе камеры используемого устройства;
3. Находиться на небольшом расстоянии от камеры используемого устройства (до 5 метров).

Объединение технологий «User Defined Targets», «User Defined Objects» и «Multi-Targets» для разработки приложения с поддержкой дополненной реальности

Приложения дополненной реальности получили широкое распространение в области мобильных приложений и видеоигр. Большинство из них имеют заранее заданные изображения-метки и не могут анализировать более одной цели одновременно, а перечень доступных AR объектов ограничен несколькими изображениями или 3D моделями. Это значит, что пользователь не может самостоятельно задать изображение-метку посредством камеры смартфона или создавать AR объекты из файлов, которые находятся непосредственно на устройстве. Перечисленные недостатки обусловлены следующими факторами:

1. Относительная новизна технологии UTD;
2. Поддержка технологии UTD только одним инструментом разработки приложений дополненной реальности – Vuforia;
3. Сложность реализации технологии UDO;
4. Высокие требования к характеристикам изображения, получаемого с камеры устройства;
5. Высокая вычислительная мощность используемого устройства.

Нами было разработано мобильное приложение, которое сочетает в себе технологии «User Defined Targets», «User Defined Objects» и «Multi-Targets». Пользователь может самостоятельно определять метки с помощью камеры мобильного устройства и создавать AR объекты, используя файлы галереи. При этом приложение способно распознавать до пяти изображений-меток одновременно.

Для разработки приложения использовался игровой движок Unity 2017 и AR библиотека Vuforia. Для реализации технологии UDO использовалась среда разработки Android Studio 2.3. На следующем рисунке представлены результаты распознавания двух изображений-меток, заданных пользователем:

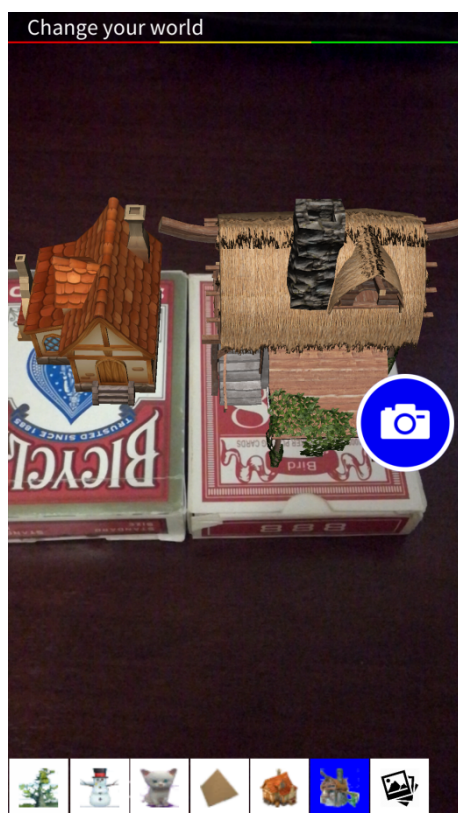


Рис. 2. Результат распознавания двух изображений-меток

Для создания пользовательского AR объекта используется специальный плагин, написанный на языке программирования Java. Разработанный плагин дает возможность получения доступа к галерее изображений Android устройства. После выбора изображения, пользователь может создать на его основе AR объект, который будет представлять собой трехмерный куб с данным представлением на его гранях. Результат создания пользовательского AR объекта представлен на рисунке 3:



Рис. 3. Результат создания AR объекта

Для корректной работы приложения следует использовать смартфон с операционной системой Android версии 5.0 и выше. Следует отметить, что приложение поддерживается всеми типами Android устройств за исключением Android TV.

Библиографический список

1. Алгоритмы распознавания маркеров дополненной реальности [Электронный ресурс], URL: <https://geektimes.ru/post/135659/> (дата обращения 08.02.18).
2. Xiang Zhang, Stephan Frons, Nassir Navab. Visual Marker Detection and Decoding in AR Systems [Текст], Darmstadt: IEEE, 2003.
3. Алан Торн. Основы анимации в Unity [Текст], М.: ДМК, 2016.
4. Крис Дикинсон. Оптимизация приложений в Unity [Текст], М.: ДМК Пресс, 2017.

УДК 004.823; ГРНТИ 28.23.13

РАЗРАБОТКА ФРЕЙМОВ ДЛЯ ОПИСАНИЯ И ВЫЯВЛЕНИЯ ОШИБОЧНЫХ ДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ ИНТЕРНЕТ-ПРИЛОЖЕНИЯ ОБРАБОТКИ КОЛИЧЕСТВЕННЫХ ЧИСЛИТЕЛЬНЫХ ЕСТЕСТВЕННЫХ ЯЗЫКОВ

А.А. Епифанцев

*Рязанский государственный радиотехнический университет,
Россия, Рязань, alexey.epifantsev@gmail.com*

Аннотация. Предложен подход к организации иерархической системы фреймов для описания ошибочных действий пользователей Интернет-приложения морфологической обработки количественных числительных. Проведена ручная классификация ошибочных ситуаций. Полученные типы ошибок объединены в группы для формального описания и программной реализации.

Ключевые слова. Фреймы, представление знаний, Интернет-приложения, морфологическая обработка, классификация.

DEVELOPMENT OF FRAMES FOR REPRESENTATION AND RECOGNIZING ERRONEOUS ACTIONS OF USERS OF THE INTERNET APPLICATION FOR PROCESSING OF QUANTITATIVE NUMERALS OF NATURAL LANGUAGES

A.A. Epifantsev

*Ryazan state radio engineering University,
Russia, Ryazan, alexey.epifantsev@gmail.com*

Abstract. We elaborate an approach of organization of the hierarchical frame system for describing the erroneous actions of users of the Internet application of processing of quantitative numerals. We classify erroneous situations manually. The obtained types of errors are grouped for formal description and program implementation.

Keywords. Frames, knowledge representation, Internet applications, morphological processing, classification.

В работе [1] предложен способ организации иерархических систем фреймов для описания действий пользователей программной системы. Способ основан на выделении базового фрейма, описывающего все допустимые явления в рамках исследуемой предметной области и последующей его конкретизации субфреймами. Конкретизация продолжается до тех пор, пока фреймы нижних уровней не будут полно и конкретно описывать возможные явления. Процесс согласования заключается в сопоставлении исследуемого явления с фреймом, находящимся как можно ниже в составленной иерархии фреймов.

Для разработки программной системы описания и выявления ошибочных действий пользователей Интернет-приложения обработки количественных числительных естественных языков необходимо сформировать структуру фреймов, основываясь на анализе статистики использования приложения [3-5].

Была проведена ручная классификация возникших в работе приложения ошибочных ситуаций. Было выделено 27 типов, которые были объединены в группы для последующего формального описания. Ниже приводится описание выделенных групп ошибок.

1. Опечатки. Под опечаткой (типографической ошибкой) принято понимать случайное отличие введенного слова от лексически корректной конструкции языка на 1-2 символа.

2. Грамматические ошибки. Это множество ошибок, причиной появления которых является недостаточное знание пользователем правил естественного языка. Сюда относятся несогласованность словоформ (*три тысяч*), порядка слов (*пять тысяч семь миллионов*), пропущенную единицу (*миллион*, правильно – *один миллион*), а также другие ошибки [1]. Эти ошибки специфичны для каждого из представленных в Интернет-приложении языков.

3. Орфографические ошибки. Это неправильное написание какого-либо слова, связанной с незнанием правил лексики естественного языка. С точки зрения программной реализации этот тип ошибок можно объединить с опечатками.

4. Указание порядкового числительного. Несмотря на то, что порядковые числительные являются корректными конструкциями естественных языков, Интернет-приложения способны обрабатывать только количественные числительные.

5. Дополнительные слова или символы. Присутствие во введенной строке слов, не имеющих отношения к количественным числительным, например, *процент, промилле, градус*, названия валют, их знаки, и т.д.

6. Указание даты или времени. Присутствие во введенной строке слов, имеющих отношение к дате и времени (названия месяцев, слова *час, минута, год*, и т.д.). Дата является частным случаем порядкового числительного.

7. Выход за диапазон. Интернет приложение способно обрабатывать количественные числительные от *MIN_VALUE* до *MAX_VALUE*. В текущей версии Интернет-приложения *MIN_VALUE* и *MAX_VALUE* равны -999999999 и 999999999 соответственно. Данное правило справедливо для всех языков.

8. Превышение допустимой точности. Интернет-приложение способно обрабатывать десятичные дроби с точностью до миллиардных.

9. Указание дроби. Входная строка представляет собой обыкновенную (правильную или неправильную) дробь.

10. Количество разделителей целой и дробной частей больше одного.

11. Неизвестная ошибка. Под неизвестной понимается ошибка, которую невозможно отнести ни к одному из вышеперечисленных типов. В контексте Интернет-приложения неизвестной ошибкой является строка, не содержащая числительных.

В результате классификации каждая из более 10 000 исследованных ошибок была отнесена к одной из перечисленных выше групп. Размеры каждой группы представлены на рисунке 1.

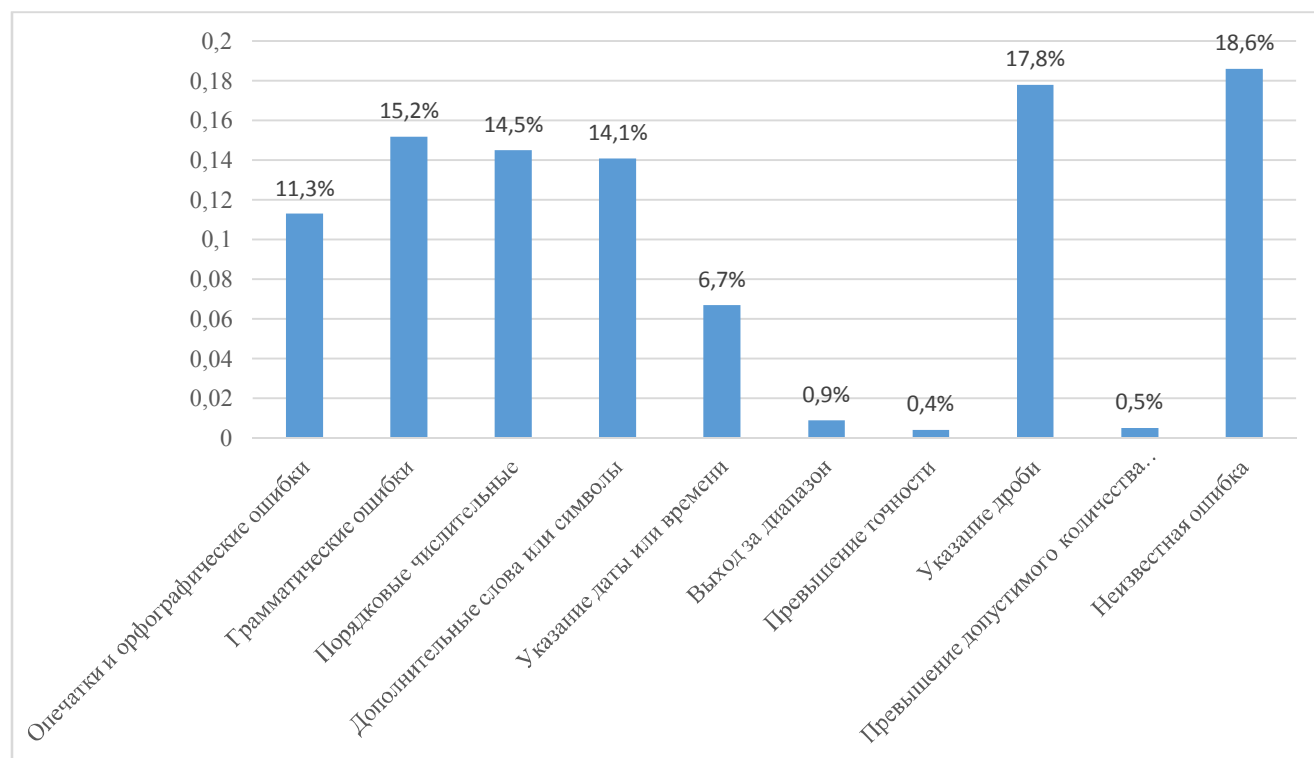


Рис. 1. Результаты классификации ошибочных ситуаций

Для заполнения хранилища фреймов программной системы необходимо формально описать каждый из вышеперечисленных типов ошибок. Для описания ошибочных действий пользователя используются фреймы, по структуре соответствующие протофрейму со следующими слотами [2]:

- название фрейма – строка;
- фрейм-родитель – указатель на другой фрейм;
- язык – элемент перечисления;
- код ошибки – строка;
- признак согласования – процедура-слуга, используется для сопоставления явления реального мира с фреймом;
- исправление ошибки – опциональная процедура-демон, запускается если исправление ошибки, описываемой фреймом, возможно.

В таблице 1 приведен пример фрейма, описывающего ошибочное действие пользователя – ввод порядкового числительного на русском языке.

Таблица 1. Описание фрейма

Имя слота	Значение слота
Название	RussianOrdinalNumberRecognizer
Фрейм-родитель	UnknownError
Язык	Русский
Код ошибки	ORDINAL_NUMBER
Признак согласования	Наличие в конце строки окончания порядковых числительных (-ый, -ое, и т.д.)
Исправление ошибки	Невозможно

Для формального описания приведенных выше групп ошибочных действий пользователей Интернет-приложения было разработано 63 фрейма. Они описывают ошибки в числительных на русском, английском, испанском, немецком, финском языках, а также записанных с помощью цифр. Разработанные фреймы объединены в иерархическую систему в соответствии с принципами [1].

Схема алгоритма согласования ошибочного действия пользователей с системой фреймов представлена на рисунке 2.

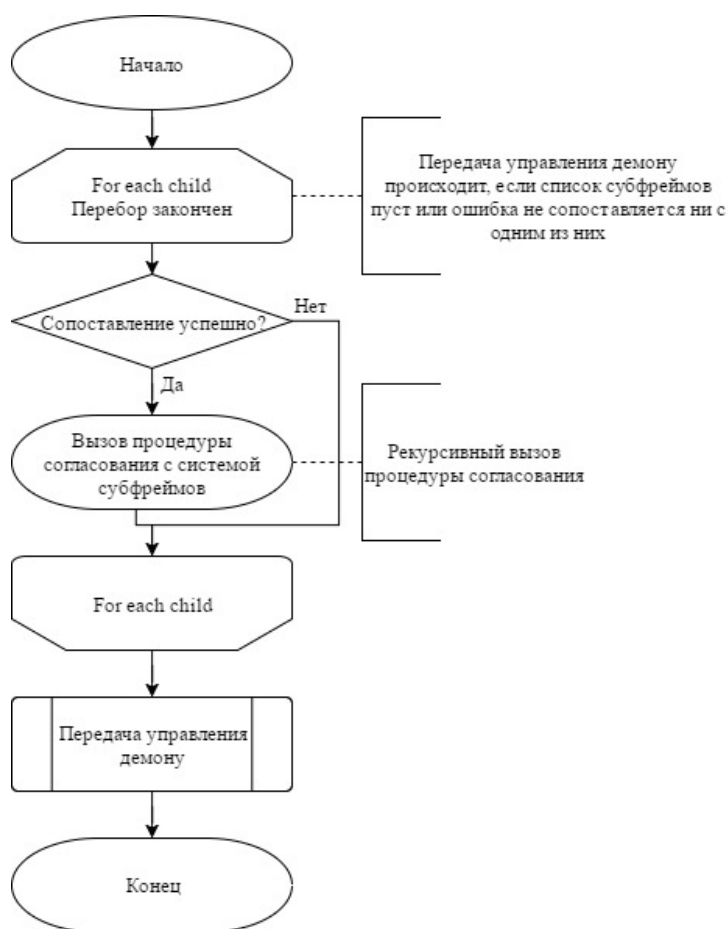


Рис. 2. Схема алгоритма согласования

Основными результатами данного исследования являются:

1. Проведена ручная классификация ошибочных ситуаций в работе Интернет-приложения.
2. Предложена структура фрейма для описания ошибочного действия пользователя Интернет-приложения.
3. Разработана иерархическая система фреймов, которая охватывает все представленные в Интернет-приложении естественные языки.
4. Разработан алгоритм согласования ошибочного действия пользователя с системой фреймов.

Главным достоинством составленной системы фреймов является способность согласовывать абсолютно любое ошибочное действие пользователя. Также структура системы позволяет расширять ее с целью конкретизации имеющихся описаний или добавления новых без изменения существующих фреймов.

Библиографический список

1. Епифанцев А.А. Способ организации иерархических систем фреймов для описания действий пользователей программной системы // Новые информационные технологии в научных исследованиях и в образовании: материалы XXI-й Всерос. науч.-техн. конф. студентов, молодых ученых и специалистов. – Рязань, 2017. – С. 138-140.
2. Минский М. Фреймы для представления знаний: пер с англ./ под ред. Ф.М. Кулакова. – М.: Энергия, 1979. – 151 с.

3. Пруцков А.В., Цыбулько Д.М. Анализ статистики использования Интернет-приложения обработки количественных числительных естественных языков // Вестник Рязанского государственного радиотехнического университета. – 2013. – № 4-1 (46). – С. 130-134.

4. Пруцков А.В., Цыбулько Д.М. Теоретико-множественное представление метода обработки количественных числительных естественных языков и особенности их перевода в различных странах // Вестник Рязанского государственного радиотехнического университета. – 2014. – № 50-2. – С. 69-75.

5. Prutzkow A. Interlingua-based numeral translation in Web-application with knowledge-testing. The European Proceedings of Social & Behavioural Sciences EPSBS. The European Proceedings of Social & Behavioural Sciences EPSBS, 2017, pp. 290-298..

УДК 004.584; ГРНТИ 20.23.21

ПРОЕКТИРОВАНИЕ ИНТЕРАКТИВНОГО ПУТЕВОДИТЕЛЯ ПО УНИВЕРСИТЕТУ

Д.Н. Карасёв, С.В. Кравченко

*Белорусский торгово-экономический университет потребительской кооперации,
Республика Беларусь, Гомель, d.krsyv@gmail.com, s_kravtchenko@tut.by*

Аннотация. В статье рассматривается работа информационных киосков и их предназначение. Описывается процесс разработки информационного киоска для Белорусского торгово-экономического университета потребительской кооперации на основе персонального компьютера с сенсорным монитором.

Ключевые слова. Информационно-справочная система, информационный киоск, университет

DESIGN OF INTERACTIVE UNIVERSITY GUIDE

D.N. Karasiou, S.V. Kravtchenko

*Belarusian Trade and Economics University of Consumer Cooperatives,
Republic of Belarus, Gomel, d.krsyv@gmail.com, s_kravtchenko@tut.by*

Abstract. The article deals with the work of interactive kiosks and their purpose. The process of developing an interactive kiosk for the Belarusian Trade and Economics University of Consumer Cooperatives on the basis of a personal computer with a touch screen is described.

Keywords. Information and reference system, interactive kiosk, university

Интерактивный путеводитель по университету – это информационно-справочная система, размещенная в здании университета, основной целью которой является обеспечение информацией о деятельности университета, его подразделениях и сотрудниках. Целевой аудиторией этой информационной системы будут абитуриенты, студенты и другие посетители университета.

Белорусский торгово-экономический университет потребительской кооперации, для которого создается интерактивная информационно-справочная система, предлагает разнообразие уровней образования и форм обучения. На образовательных программах университета обучаются представители различных поколений – от пожилых людей в Институте третьего возраста до младших школьников и даже детей дошкольного возраста на курсах иностранных языков. Университет предлагает обучение на I и II ступенях получения образования на очной и заочной формах, в аспирантуре, на обучающих курсах и факультете повышения квалификации и переподготовке. Кроме образовательных услуг университет оказывает и другие платные услуги: консалтинговые, полиграфические, транспортные, услуги общепита. Таким образом, посетителями университета могут быть различные лица с самыми различными целями наведывания университетского комплекса.

Создание информационно-поисковых систем и информирование пользователей – не новая задача. Она реализуется в виде специализированных приложений или программно-технических устройств, называемых информационными киосками.

Информационный киоск, или инфокиоск - это устройство, созданное на основе компьютера, обычно с мультимедийным дисплеем, стационарно установленное в местах общего пользования для обеспечения широкой общественности доступа к информации. Инфокиоск – это устройство самообслуживания, которое позволяет пользователю самостоятельно получать информацию или выполнять сервисные операции, не прибегая к помощи сотрудников организации-владельца инфокиоска. Размещаются такие устройства в общественных местах или местах с большим количеством посетителей или клиентов, таких как торговые центры. Информационный киоск может быть дополнительно оснащен клавиатурой, купюроприемником, термопринтером, сканером штрих-кодов и другим оборудованием.

Сфера применения таких устройств как информационные киоски обширна. В банковском секторе через инфокиоск можно оплачивать коммунальные услуги и осуществлять другие платежи, проверить баланс карт-счета, получить справочную информацию о продуктах и услугах банка. Сенсорные киоски используются в торговых центрах или аэропортах для ориентирования посетителей, показывая подробный план здания и давая наглядные инструкции, как добраться до нужного места. В крупном супермаркете киоск информирует о наличии товаров и их местонахождении. Справочный терминал может сообщить подробную информацию об интересующем товаре или услуге, и в этом случае он заменяет собой бумажный каталог фирмы. В учебном заведении установленный терминал может предоставить расписание занятий в удобном электронном виде. Музейную или художественную экспозицию инфокиоск может расширить дополнительными текстовыми, графическими, фото- или видео- материалами, информацией о будущих выставках. Установленный на вокзале или кинотеатре информационный киоск может продать билет. В кафе и ресторане инфокиоск заменяет собой официанта, так как позволяет выбрать блюда, составить заказ и даже оплатить его банковской картой. [1]

Таким образом, информационный киоск – устройство, повышающее степень комфорта покупателей или посетителей, а, следовательно, и степень их удовлетворенности.

В Беларуси, в основном, информационные киоски являются банковскими. Следующей по популярности сферой их применения является организация электронной очереди в тех местах, где наблюдается большой наплыв посетителей. Это могут быть почтовые отделения, отделения банков, крупные аптеки. В этом случае, кроме инфокиоска, устанавливаются информационные панно, высвечивающие номер очереди и оператора, который обслужит клиента.

В Республике Беларусь производителем сенсорных информационных киосков является ЗАО «НПП БЕЛСОФТ» [2]. Компания производит широкий спектр инфокиосков: плоские напольные (с чековым термопринтером, клавиатурой и без), плоские крупноформатные (с чековым термопринтером и без), настенные. Существует также возможность закупки сенсорных киосков зарубежного производства.

Несмотря на то, что на рынке справочных терминалов предлагаются решения различного ценового диапазона, в том числе и недорогие, университет не планирует приобретать информационный киоск даже в одном экземпляре. Было принято решение заменить данное устройство обычным сенсорным монитором, подключенным к стационарному ПК. В качестве сенсорной панели был взят монитор компании Samsung LS24C770 с диагональю в 24 дюйма, уже приобретенный ранее и имеющийся в наличии в университете. Программной начинкой было принято решение написать компьютерное приложение, напоминающее работу пользователя с обычным информационным киоском.

Целевой аудиторией, для которой создается приложение интерактивного путеводителя по университету, являются обучающиеся, сотрудники и все посетители высшего учебного заведения.

При создании информационного киоска для университета следует принять проектные решения по следующим вопросам:

- место размещения устройства;
- контент и интерфейс разрабатываемого приложения.

Проблема выбора места для физического размещения интерактивного справочного киоска является основной, а не второстепенной, наряду с его содержательным наполнением. Рекомендуемым местом для инфокиоска является место с высокой посещаемостью или проходимостью, что выявляется анализом направленности потоков посетителей учебного заведения.

Так как планируется разместить единственный сенсорный киоск, то следует выбрать общедоступное помещение с наибольшим потоком потенциальных пользователей. Таким помещением является холл первого этажа главного учебного корпуса университета. Схема холла показана на рисунке 1. Холл, где будет помещен инфокиоск, находится сразу за главным входом в университет, потому желательно, чтобы киоск находился рядом с входом, что и показано на схеме (рис. 1). Это позволит посетителям университета, впервые попавшим в его стены, быстрее отыскать инфокиоск и с его помощью легко сориентироваться в незнакомом месте. Несмотря на то, что инфокиоск разместится в самом многолюдном месте университета – холле первого этажа – его место выбрано удачно, чтобы не мешать большим людским потокам, например, студентам, спешащим к первой паре.

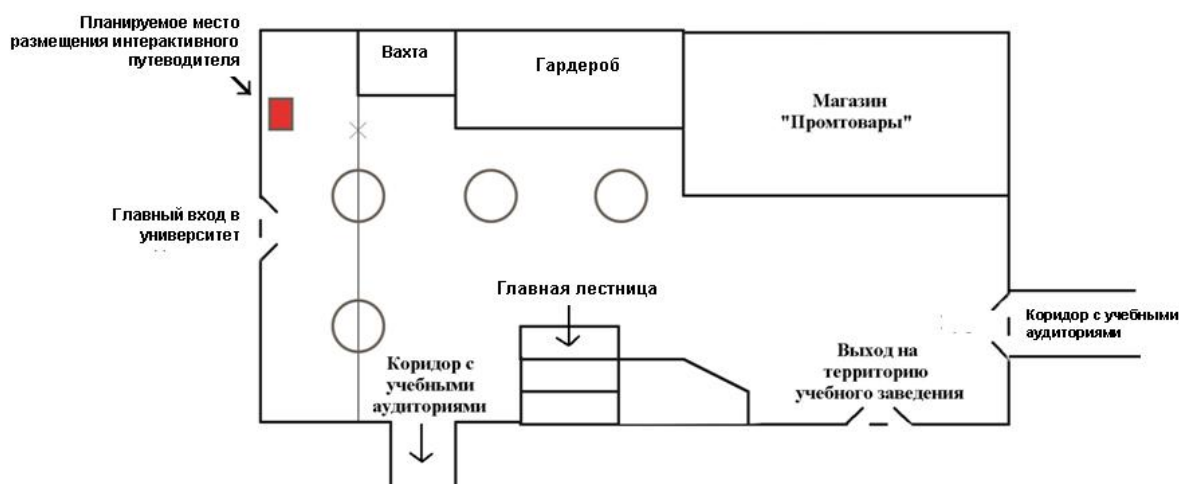


Рис. 1. Холл первого этажа главного корпуса университета

Правильно подобранный контент, эстетический и функциональный дизайн приложения являются ключевыми моментами для пользователей. Ввиду того, что инфокиоск в холле университета будут использовать разные группы пользователей с различными навыками работы на компьютере, то интерфейс интерактивного путеводителя должен быть интуитивно понятным и простым. В интерфейсе информационного киоска рекомендуется использовать крупные кнопки, небольшое количество возможностей ветвления и ограниченное количество страниц, которые могут быть вызваны. Приложение для инфокиоска – это информационная система, ограниченная по иерархической глубине и сложности. Дизайн интерфейса разрабатывается, как правило, на основании корпоративных цветов организации. Для университета корпоративным цветом является синий (рис. 2).



Рис. 2. Главный (первый) экран интерактивного путеводителя

Главным средством коммуникации с широкой аудиторией для университета является его веб-сайт. Он содержит подробные сведения о разнообразных направлениях деятельности вуза. Интерактивный путеводитель предназначен для тех, кто не может зайти на сайт университета и получить исчерпывающую информацию. При этом информационно-справочная система инфокиоска не должна дублировать сайт не по структуре, ни по содержанию. В качестве языка для реализации интерфейса выбран русский язык – один из двух государственных языков Республики Беларусь. Дублировать содержание информационной системы на другом языке, например, английском, не планируется.

Уходят в прошлое времена, когда посетители университета обращались к вахтеру у главного входа с вопросом: «Как пройти в ...?». Вахтер не всегда владеет нужной информацией, это не входит в его обязанности. Полные сведения предоставит веб-сайт, и как дополнение к нему – сенсорный информационный киоск. Интерактивная система-путеводитель, встречающая гостей вуза, будет способствовать имиджу университета как передового, идущего в ногу с развитием информационно-коммуникационных технологий.

Библиографический список

1. Найдется в киоске. Интерактивный сенсорный киоск как информатор о товаре, электронный гид, справочная и рекламная площадка. [Электронный ресурс]. – Режим доступа: http://touch.ru/articles/interaktivnyy_sensornyy_kiosk_kak_informator_o_tovare_elektronnyy_gid_spravochnaya_i_reklamnaya_ploshchadka/ – Дата доступа: 11.02.2018.
2. Информационные киоски Белсофт [Электронный ресурс]. – Режим доступа: <http://www.belsoft.by/site/ru/products/manufacture/infokioski>. – Дата доступа: 11.02.2018.

УДК 004.823; ГРНТИ 28.23.13

ПРОВЕРКА КАЧЕСТВА КОДА ПРИ АВТОМАТИЗИРОВАННОМ СОСТАВЛЕНИИ ОТЧЕТОВ ПО ПРАКТИЧЕСКИМ ЗАДАНИЯМ

А.И. Николаев

Рязанский государственный радиотехнический университет,

РФ, г.Рязань, andrey.nikolaev130@gmail.com

Аннотация. В данной работе рассматриваются средства автоматической проверки качества кода и их интеграция в разрабатываемую программную систему автоматизированного составления отчетов по практическим заданиям.

Ключевые слова. Качество кода, статический анализатор, соглашение по написанию кода, автоматизированное составление отчетов.

CODE QUALITY CHECK IN AUTOMATED REPORTING SYSTEM FOR PRACTICAL WORKS

A.I. Nikolaev

Ryazan State Radio Engineering University,

Russian Federation, Ryazan, andrey.nikolaev130@gmail.com

Abstract. We discuss tools for automatically checking code quality and integrating it into a developing software system for automatic practical work reporting.

Keywords. Code quality, static analyzer, code conventions, automated reporting.

В программной системе автоматизированного составления отчетов по практическим заданиям[1] кроме составления отчетов по заданному шаблону требуется проверять качество написанного кода. Это упростит проверку практического задания преподавателю, а также позволит студенту сразу увидеть ошибки, сделанные в коде. Студент сможет их исправить, до того как покажет выполненное задание преподавателю. В результате сдача практического задания будет занимать меньше времени.

Главным критерием при написании кода является его работоспособность. Но, кроме того, что код должен просто работать, часто к нему могут предъявляться дополнительные требования: читаемость, управляемость, соответствие определенным стандартам, наличие комментариев. То есть то, что можно назвать качеством кода.

И если работоспособность кода мы можем определить с помощью проверочных тестов, то как быть с остальными параметрами? Под качеством кода понимается набор субъективных оценок восприятия кода другим человеком. Студент может получить отзыв о качестве своего кода от преподавателя, но преподаватель – это, прежде всего, человек, и он может ошибаться, пропускать некоторые участки некачественного кода. Более того, анализ кода каждого студента занимает длительное время. Поэтому необходимо найти автоматический способ выполнения этой задачи.

Требования, которые могут предъявляться к коду при выполнении практических заданий по программированию:

- компилируемость всего кода;
- работоспособность кода;
- читаемость отдельных алгоритмов, методов, строк;
- управляемость кода;
- соответствие соглашению по написанию кода;
- наличие комментариев для каждого участка кода.

Отличие между компиляцией кода и его работоспособностью в том, что если код компилируется, это еще не значит что он работоспособный и решает поставленную задачу. Для проверки работы кода, повсеместно используются автоматические тесты.

При оценке читаемости кода, рассматривается возможность, глядя на код, быстро понять реализованный алгоритм, и оценить, как будет вести себя программа в том или ином частном случае.

Под управляемостью понимается возможность в минимальные сроки внести в код требуемые изменения, избежав различных неприятных последствий.

Для автоматической проверки качества кода можно использовать следующие средства: Checkstyle[2], FindBugs[3], PMD[4].

Checkstyle – это программа статического анализа исходного кода, используемая при разработке программного обеспечения для проверки соответствия правилам написания исходного Java-кода[5]. Этот инструмент автоматизирует процесс проверки кода и отстраняет людей от участия в этой задаче. Это средство работает с исходным кодом программы и строит по нему абстрактное синтаксическое дерево, которое в последствии и будет анализироваться. Файл настроек представляет собой xml файл, в котором указывается какие проверки необходимо выполнять. По умолчанию этот инструмент поставляется с двумя конфигурациями: Sun Code Conventions и Google Java Style, но мы можем с легкостью создать свой файл с настройками. Примеры некоторых наиболее часто используемых проверок: размер класса не более N строк, в конструкции try-catch, блок catch не пустой, наличие комментариев, порядок членов в классе, порядок модификаторов переменных, наличие магических чисел, обязательность {} в if, наличие ветки по умолчанию в switch, парность и ковариантность hashCode и equals. Всего содержится более 150 различных проверок, которые позволяют нам задать определенный стиль написания кода. Но при необходимости мы в любой момент можем реализовать недостающие нам проверки сами, так как это средство имеет открытый исходный код и открыто к изменению.

При интеграции этого инструмента в программную систему автоматизированного составления отчетов по практическим заданиям возникли некоторые проблемы. Checkstyle обычно поставляется в виде плагина для среды разработки или в качестве отдельного приложения, а в качестве отдельной библиотеки его нет. Поэтому пришлось написать свою библиотеку, которая включила в себя исходный код Checkstyle и интерфейс взаимодействия с ним, который позволил нам использовать все возможности этого инструмента в других проектах.

FindBugs – это статический анализатор кода с открытым исходным кодом, который находит проблемы и недостатки в программах Java и это сильно помогает улучшить качество кода. В отличие от Checkstyle, этот инструмент анализирует байт-код программы. Это средство знает о более чем 400 шаблонах ошибок и каждый из шаблонов сгруппирован по некоторым группам: производительность, ужасный код, безопасность, корректность, многопоточная правильность, плохая практика. Так же каждому шаблону присваивается приоритет: максимальный, средний, низкий. Рассмотрим некоторые примеры для более хорошего понимания.

Пример ошибки, которая относится к правильности кода:

```
String name = "Andrey";  
name.toUpperCase();  
if (name.equals("ANDREY"));
```

Возвращаемое значение методов не должно быть игнорировано. В Java строка – это неизменяемый объект, поэтому когда мы вызываем метод toUpper(), то он не меняет нашу строку name, и следовательно, наше условие будет false. Для правильного выполнения необходимо переопределить значение name = name.toUpperCase(). Поэтому игнорирование возвращаемого значения метода будет считаться ошибкой и FindBugs нам об этом сообщит.

Пример ошибки производительности

```
for (int x = 0; x < exceptions.size(); x++) {  
    errorMessage += getStackTrace(  
        exceptions.get(x) + "\n");  
}
```

В данном коде происходит объединение строки с использованием +. Как известно это довольно медленная операция и лучше использовать для этого StringBuilder или StringBuffer. Об этом улучшении FindBugs тут же вам сообщит.

Пример возможного указателя на null

```
Person person = aMap.get("bob");  
String name = person.getName();
```

В этом примере aMap может не содержать человека с именем "bob", и тогда нам вернется null. И в следующей строке мы получим исключение NullPointerException, когда попробуем узнать имя человека. Так как FindBugs не знает о том, есть ли в aMap человек с именем "bob", то он пометит вторую строчку специальным флагом, который означает, что в этой строке может вызваться исключение указателя на null. Вручную места, которые могут вызвать данное исключение отследить очень сложно даже опытному программисту.

Мы рассмотрели некоторые примеры ошибок, которые может найти FindBugs. Это средство очень полезно использовать тем, кто только знакомится с языком, чтобы узнать об особенностях языка и быстрее в нем разобраться. Кроме того оно также полезно и опытным программистам, чтобы найти ошибки, которые они допустили в своем коде по невнимательности.

PMD – это расширяемый многоуровневый статический анализатор кода. Он так же как и FindBugs находит ошибки в коде, но он анализирует исходный код. Выделяются следующие пять типов ошибок: критическая ошибка, ошибка, критическое предупреждение, предупреждение, информативное предупреждение. Некоторые ошибки дублируются с FindBugs, но главное преимущество PMD в том, что он может вам показать цикломатическую сложность программы. Так же данное средство анализа показывает нам участки с дублированным кодом.

При интеграции PMD и FindBugs в программную систему автоматизированного составления отчетов, возникают те же проблемы, что и с Checkstyle: все эти средства поставляются в виде плагинов для сред разработки или в виде отдельных приложений, но их нет в виде библиотек.

Все вышеперечисленные выше средства анализа качества кода следует использовать вместе, так как они дополняют друг друга. Кроме того, они помогают быстрее освоить новый язык программирования и узнать о его особенностях. Так что эти средства прекрасно подходят для обучения студентов и их необходимо использовать при выполнении практических заданий. Эти программы могут быть использованы как плагины для среды разработки.

Рассмотренное средство проверки качества написания кода Checkstyle уже внедрено в систему автоматизированного составления отчетов по практическим заданиям. Эта система будет использована для повышения качества учебного процесса и будет внедрена в дисциплину «Программирование на Java» [6-7] и другие дисциплины кафедры «Вычислительная и прикладная математика» РГРТУ, в которых преподается язык программирования Java, для помощи преподавателю при проверке практических заданий и студентам при их выполнении и составлении отчетов по ним. Дальнейшая разработка может быть направлена на добавление в данную систему других средств проверки качества кода, таких как: FindBugs и PMD.

Программная система разрабатывается на языке Java в рамках выпускной квалификационной работы.

Библиографический список

1. Николаев А.И. Программная система автоматизированного составления отчетов по лабораторным работам // Новые информационные технологии в научных исследованиях – НИТ-2017: материалы XXII Всероссийск. научн.-техн. конф. студентов, молодых ученых и специалистов./Рязан.гос.радиотехн.ун-т. – Рязань, 2017. – С. 184-185.
2. Checkstyle [Электронный ресурс], URL: <http://checkstyle.sourceforge.net/> (дата обращения 13.02.18)
3. FindBugs [Электронный ресурс], URL: <http://findbugs.sourceforge.net/> (дата обращения 13.02.18)
4. PMD [Электронный ресурс], URL : <https://pmd.github.io/> (дата обращения 13.02.18)
5. Блинов И.Н., Романчик В.С. Java. Методы программирования: учеб.-метод. пособие. – Минск:Изд-во «Четыре четверти», 2013. – 896 с.
6. Пруцков А.В. Особенности преподавания промышленной разработки программных продуктов в технических вузах // Современные технологии в науке и образовании – СТНО-2017: материалы 2-й Междунар. науч.-техн. и науч.-метод. конф. / под общ. ред. О.В. Миловзорова / Рязан. гос. радиотехн. ун-т. – Рязань, 2017. – С. 40-41.
7. Пруцков А.В. Программирование на языке Java. Введение в курс с примерами и практическими заданиями: учебник. – М.: КУРС, 2018. – 208 с.

УДК 004.823; ГРНТИ 28.23.13

ДИВЕРГЕНТНОЕ КОДИРОВАНИЕ С АЛГОРИТМОМ ВИТЕРБИ ОКОЛО ГРАНИЦЫ ШЕННОНА

В.В. Золотарёв*, Г.В. Овечкин, В.В. Корябкин****

**Институт космических исследований РАН, Россия, Москва*

*** Рязанский государственный радиотехнический университет,
Россия, Рязань, koryabkin.vlad@gmail.com*

Аннотация. Рассмотрены методы реализации принципа дивергенции с использованием для декодирования одновременно двух типов декодеров: алгоритма Витерби (АВ) и многопороговых декодеров (МПД). Указаны преимущества этих алгоритмов, относящихся к группе декодеров с прямым контролем метрики (ДПКМ). Рассмотрены подходы к реализации мультиалгоритмического декодирования, которые могут существенно снизить сложность декодирования вблизи границы Шеннона.

Ключевые слова. Многопороговое декодирование, дивергентное кодирование, каскадирование, коды контроля по чётности, граница Шеннона, алгоритм Витерби, пропускная способность канала, сложность декодирования, поиск глобального экстремума, оптимальное декодирование, кодовая скорость, энергетический выигрыш кодирования, оптимизационные алгоритмы.

DIVERGENT CODING WITH VITERBY ALGORITHM NEAR THE SHENNON BORDER

V.V. Zolotarev*, G.V. Ovechkin, V.V. Koryabkin****

** Space Research Institute of the Russian Academy of Sciences, Russia, Moscow*

*** Ryazan state radio engineering university, Russia, Ryazan, koryabkin.vlad@gmail.com*

Abstract. The methods for implementing the divergence principle are considered with the use of two types of decoders for decoding simultaneously: the Viterbi algorithm (AV) and the multithreshold decoders (MTD). Advantages of these algorithms relating to the group of decoders with direct metric control (DPCM) are indicated. The approaches to the implementation of multi-algorithmic decoding, which can significantly reduce the complexity of decoding near the Shannon boundary, are considered.

Keywords. Multithreshold decoding, divergent coding, cascading, parity check codes, Shannon boundary, Viterbi algorithm, channel capacity, decoding complexity, for example global extremum, optimal decoding, code rate, power gain of coding, optimization algorithms.

Как известно, на основе Оптимизационной Теории (ОТ) возможна реализация многопороговых декодеров МПД, которые используют методы поиска глобального экстремума функционалов при линейной от длины кода сложности декодирования. Такие алгоритмы обеспечивают высокий уровень результирующей достоверности декодирования, во многих случаях практически совпадающей с результатами оптимального декодирования (ОД) непосредственно вблизи границы Шеннона, т. е. около пропускной способности канала [1-3]. Важным условием работоспособности МПД является применение мажоритарно декодируемых кодов с малым уровнем размножения ошибок (РО), правильный выбор типа самого алгоритма МПД и оптимальная настройка элементов декодера.

Большое значение при создании алгоритмов МПД, работающих вблизи границы Шеннона, имеет правильное применение соответствующих ОТ парадигм (совокупности правил, методик, закономерностей и свойств), определяющих конечную эффективность методов ОТ и создаваемых на её основе алгоритмов МПД и других успешно взаимодействующих с ними методов. Ниже рассмотрены характеристики новых вариантов коррекции ошибок, при создании которых оказалось важным правильно применять принципы дивергентного кодирования [3-5]. Мы сосредоточим далее наше внимание на совместном применении для этого АВ и МПД методов. Важнейшей причиной такого выбора анализируемых алгоритмов является их принадлежность группе декодеров с прямым контролем метрики (ДПКМ). По нашим представлениям, в настоящий момент только все модификации АВ [9,10] и МПД алгоритмов, включая символьные декодеры [1,7,8,11-13], обладают важнейшим свойством точного измерения расстояния своих решений до принятого сообщения, которое и должен минимизировать ОД, например, на основе АВ, или метод поиска глобального экстремума, единственными представителями которого пока являются МПД декодеры. Все другие методы коррекции ошибок или вообще "не озабочены" оценкой этого расстояния, как, в частности, работают алгебраические процедуры, или как бы измеряют некоторые "функции от функции", связанные указанными расстояниями, да и только! К последним относятся уже давно известные и остановившиеся в своём развитии низкоплотностные (LDPC) и турбо коды. Их достаточно эффективная работа при умеренно большом уровне шума в канале обусловлена тем, что они реализуют очень затратные при реализации методы, гораздо более сложные, чем МПД или каскадные методы на основе МПД и АВ.

Принцип дивергентности (расширения, увеличения) в ОТ состоит в реализации различных методов увеличения кодового расстояния некаскадными методами, что позволяет повысить работоспособность МПД декодеров при большом уровне шума [3-5]. Особая простота реализации алгоритмов на основе ОТ иллюстрируется демопрограммами, которые можно переписать с наших сетевых ресурсов [4]. Например, с их учебных страниц можно взять сверхбыструю демопрограмму по МПД алгоритму по гиперссылкам (demo_quick.zip) и инструкцию [4] к ней, а также с главных страниц "О методе" этих порталов - демонстрационную программу со своим "описанием" [4]. Первая из этих демопрограмм показывает быструю, на скорости работы более 10Мбит/с, работу программного МПД декодера в гауссовском канале с большим уровнем шума, а вторая показывает мультфильм о процессе сходимости решений МПД при большом шуме канала к решению ОД на основе глобального поиска экстремума (минимума!) функционала.

На рисунке 1 представлена простая схема из [5], иллюстрирующая основные идеи дивергенции [1,3], впервые изложенные тоже в [5].

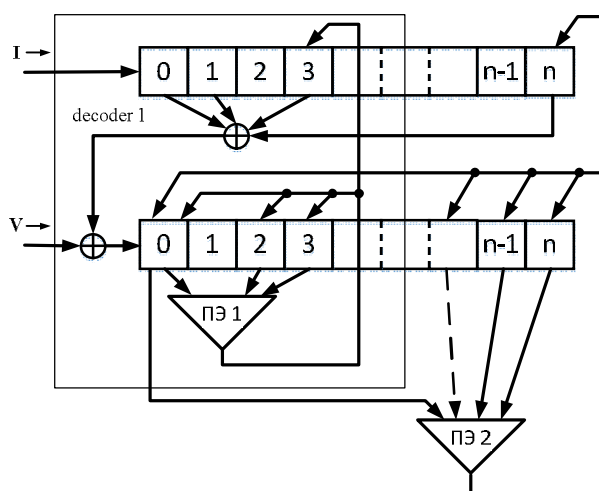


Рис.1. Иллюстрация принципа дивергенции

Представленная схема при всей своей простоте отображает очень сложный механизм взаимодействия двух алгоритмов, более детально изложенный в [1]. Фактически весьма простой вариант МПД декодера [1,4,6-8] устроен на рис.1 так, что области регистра синдрома, с которых снимаются сигналы на ПЭ, пересекаются. Отметим, что не представленный на схеме разностный регистр в МПД не указан просто для упрощения рисунка, поскольку он не несёт основной смысловой нагрузки в идее дивергенции. А пересечение синдромных областей позволяет алгоритму МПД приобрести новое свойство, которое состоит в том, что первый ПЭ1 снижает плотность ошибок перед вторым ПЭ2. Это позволяет второму ПЭ2 значительно уменьшить в свою очередь и оставшуюся после него долю ошибок по сравнению со случаем, если бы первого ПЭ1 не было бы вообще. Но положение осложняется тем, что на входе ПЭ1 на одну проверку меньше, чем у ПЭ2, что усложняет его работу, поскольку ошибки, присутствующие в этой проверке, только увеличивают уровень шума на входе ПЭ1, а использовать это проверку он не может. Но, как оказывается, при правильном и весьма тонком выборе параметров МПД, представленного на рис.1, можно реально повысить уровень шума, при котором такой МПД будет работать успешно. Более того этот подход можно применить несколько раз. Тогда по мере увеличения длины регистров МПД и числа ПЭ, расположенных на них, количество используемых ими проверок будет расти и каждый такой ПЭ будет работать со своим кодом со всё большим кодовым расстоянием. Обратим внимание на то, что после почти 50-летнего развития теории ОТ и МПД все очевидные и простые методы улучшения характеристик МПД уже найдены и использованы. Поэтому выделение в самостоятельный метод столь мощной идеи дивергенции является очень значимым шагом развития ОТ.

А теперь обратим внимание на то, что вместо первого ПЭ1 в представленной схеме декодирования может быть любое решающее устройство, т. е. перед ПЭ2 можно поставить (и не раз!) какой-то другой алгоритм коррекции ошибок. А с учётом сделанных выше комментариев, можно согласиться с тем, что в качестве наиболее предпочтительного декодера очень желательно попробовать применить именно АВ, который входит в группу ДПКМ и поэтому может быть лучше и проще других согласован с МПД декодерами.

На рисунке 2 представлены графики, соответствующие возможностям АВ и некоторым другим методам коррекции ошибок в АБГШ канале при $R=1/2$: 1 – АВ для сверточного кода с $K=24$, который пока очень сложен для реализации; 2 – АВ для кода с $K=15$; 3 – АВ для кода с $K=7$, 4 – характеристики ранее рассмотренного первого МПД (кривая 1 на рис. 3); 5 – каскадная схема, состоящая из сверточного кода с $K=7$ и кода Рида-Соломона.

Обратим далее внимание на то, что при замене на рис. 1 первого ПЭ1 на декодер Витерби для систематического кода при достаточной длине информационного регистра декодера с крайней n -ой ячейки этого регистра в регистр проверочных символов такого составного декодера будут поступать ошибки декодера Витерби, что будет дополнительным шумом на его входе. Из-за этой положительной обратной связи характеристики АВ несколько ухудшатся, что еще несколько повысит вероятность ошибки АВ. Однако при правильном выборе уровня шума канала воздействие дополнительного шума будет небольшим.

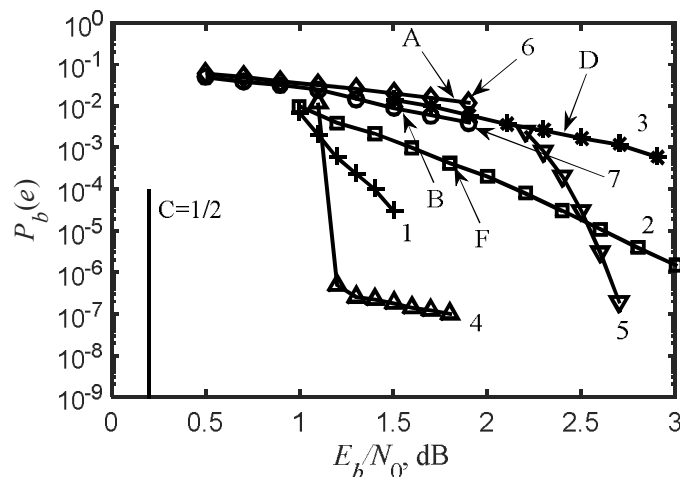


Рис. 2. Характеристики дивергентных МПД с АВ

Оценим энергетику канала, при которой будет работоспособен декодер нового типа в схеме АВ и следующего за ним МПД, работающий с кодом, у которого кодовое расстояние d на единицу больше, чем у кода для АВ, который стоит перед ним. Понятно, что хорошим ориентиром здесь будет такой уровень шума в канале, когда при некоторой вероятности ошибок от n -й ячейки на выходе АВ эта вероятность еще будет несколько меньшей. Данный параметр удобно использовать для оценки эффективности такой схемы, поскольку из n -й ячейки на вход АВ идут именно оставшиеся ошибки самого АВ. Нужно только учитывать, что в реальной схеме такого комплексного декодера ошибки АВ сильно группируются. Поэтому перед МПД декодером нужно поставить перемежитель, чтобы ошибки на вход АВ приходили в случайном порядке.

Выбрав схему оценок, сначала рассмотрим варианты взаимодействия АВ и МПД для систематических кодов. График 6 показывает характеристики АВ для систематического кода с $K=7$. Стрелка A указывает уровень шума канала, ниже которого будет возможность поставить МПД, который снизит вероятность ошибки декодирования. Но увеличение кодового расстояния лишь на 1 по сравнению с АВ не очень сильно улучшит характеристики такого комплексного декодера. Для улучшения возможностей такой схемы нужно еще удлинить регистры декодера (конечно, и кодера!), на которых также и с других информационных ячеек снимаются значения, также направляемые на вход декодера Витерби. Иначе говоря, нужно многократным применением дивергенции значительно увеличить без какого-либо каскадирования кодовое расстояние, что при его значительном увеличении и росте числа дополнительных ПЭ в МПД декодере позволит в конце концов уже на много порядков понизить вероятности ошибки рассматриваемого нового декодера дивергентного типа. Кроме того, нужно учесть, что увеличение числа проверок, с выходов которых поступают ошибки на вход АВ, также снижает допустимую энергетику канала, при которой эта схема будет работать. Конечно, вероятности ошибок со все более далеких ячеек достаточно быстро уменьшаются,

поскольку большое число ПЭ будет последовательно быстро снижать число оставшихся ошибок после них, именно тех, которые и поступают одновременно на вход АВ, в его проверочный регистр. Таким образом, итоговые возможности предлагаемой схемы нужно получать при моделировании этой схемы и подборе параметров кодов и декодеров.

Рассмотрим далее возможности АВ для систематического кода с $K=15$. Характеристики его представлены графиком 7, а найденная по тем же критериям граница рабочей области обсуждаемого типа декодеров указана стрелкой *B*. Естественно, что более сложный АВ обеспечит работу комплексного декодера при уровне шума почти на 0.5 дБ больше. Но более точное определение границ рабочей области нового типа декодера и здесь следует делать на основе моделирования.

Хорошо известно, что несистематические сверточные коды имеют гораздо большие возможные значения минимального кодового расстояния, чем систематические. А это в свою очередь обеспечивает и существенно меньшие вероятности ошибки, что активно используется при создании декодеров на основе АВ. При этом известно, что несистематические МПД реализуются столь же легко, как и обычные систематические [7, 8].

Аналогичные методы оценки возможностей предлагаемого нового метода для несистематических кодов показали для кодов с $K=7$ и $K=15$, что границы эффективности комплексных декодеров указаны стрелкам *D* и *F*. Как можно заключить из сопоставления применения кодов, в связи с меньшим уровнем группирования ошибок в систематических кодах, их применение в рассмотренных схемах более предпочтительно из-за меньшей допустимой энергии канала. Однако постановка задачи и сделанные предварительные оценки возможности совместной работы АВ и МПД должны быть на следующем этапе исследования существенно уточнены при полном моделировании работы такой схемы.

Весьма существенно также и то, что область эффективной работы предлагаемой некаскадной схемы может находиться в области гораздо больших шумов, чем у все еще популярной каскадной схемы с кодами Рида-Соломона, также представленной на рис. 2. Это момент будет уточнен, конечно, только после проведения полного моделирования работы предложенной схемы.

На самом деле эта задача с неизбежностью требует еще одного шага исследования для получения, возможно, самого технологичного варианта декодирования. Применение на первом этапе коррекции в этой схеме алгоритма Витерби определяется, конечно же, его оптимальностью. Но, хотя даже АВ с длиной используемого кода $K=15$ – вполне реализуемый вариант декодера, он оказывается достаточно сложен по сравнению с МПД. Поэтому после проведения анализа работы такой схемы декодера для различных кодов целесообразно найти вместо АВ для первого этапа декодирования какой-либо алгоритм с меньшей сложностью, чем АВ, но с достаточно высокой эффективностью предварительного декодирования. Это упростит итоговый декодер и существенно ускорит его работу, возможно, при сохранении почти таких же характеристик декодирования, как и у схемы с алгоритмом Витерби.

Заключение

В докладе представлены основные результаты разработки простых для реализации и эффективных дивергентных методов декодирования помехоустойчивых кодов, которые могут быть использованы в телекоммуникационных системах. Основой разработанных методов коррекции ошибок являлись многопороговые алгоритмы декодирования самоортогональных кодов. Применение предложенных в работе методов позволяет обеспечить эффективное декодирование в непосредственной близости от пропускной способности канала при малой сложности реализации декодера. Предложенные в работе идеи для двоичных кодов легко обобщаются на 3D каскадирование символьных кодов.

Работа выполнена при поддержке Российского фонда фундаментальных исследований.

Библиографический список

1. В.В. Золотарёв. Теория кодирования как задача поиска глобального экстремума. - Под научной редакцией академика РАН Н.А. Кузнецова- // М., "Горячая линия - Телеком", 2018, 220 с. (в печати).
2. Zolotarev V.V., Ovechkin G.V., Chulkov I.V., Ovechkin P.V., Averin S.V., Satibaldina D.Zh., Kao V.T. Review of achievements in the optimization coding theory for satellite channels and Earth remote sensing systems: 25 years of evolution, *Sovremennye Problemy Distsionnogo Zondirovaniya Zemli iz Kosmosa*, 2017, vol.14, no.1, pp.9-24.
3. В.В. Золотарёв. Способ декодирования информации с использованием сверточных кодов (принцип дивергенции) // Патент на изобретение РФ №2637487 от 04.12.2017 г.
4. Ресурсы www.mtdbest.iki.rssi.ru и www.mtdbest.ru.
5. Золотарёв В.В., Овечкин Г.В. Дивергентное кодирование свёрточных кодов // Материалы 18-й Международной научно-технической конференции "Проблемы передачи и обработки информации в сетях и системах телекоммуникаций", 2015, с. 27–32.
6. В.В. Золотарёв, Г.В. Овечкин, П.В. Овечкин. Эффективные многопороговые методы декодирования самоортogonalных кодов. //«Вестник РГРТУ», №60,2017г., с.113-122.
7. В.В. Золотарёв, Ю.Б. Зубарев, Г.В. Овечкин. Многопороговые декодеры и оптимизационная теория кодирования. // Под редакцией академика РАН В.К. Левина. М., «Горячая линия – Телеком», 2012, 238с.
8. V.V. Zolotarev, Y.B. Zubarev, G.V. Ovechkin. Optimization Coding Theory and Multithreshold Algorithms. // Geneva, ITU, 2015, 159p.
9. В.В. Золотарёв, П.В. Овечкин. Способ кодирования и декодирования блочного кода с использованием алгоритма Витерби. // Патент РФ №2608872 от 25.01.2017 г.
10. В.В. Золотарёв, Г.В. Овечкин, П.В. Овечкин. Характеристики блочных реализаций алгоритма Витерби. // «Вестник РГРТУ», №59, 2017г.
11. В.В. Золотарёв. Патент РФ на изобретение №2557454 "Способ декодирования помехоустойчивого кода" от 25.06.2015.
12. В.В. Золотарёв, Д.Ж. Сатыбалдина, А.Д. Адамова. Метод ускорения работы символьного порогового элемента многопорогового декодера. // Вестник Евразийского национального университета им. Л.Н. Гумилёва (РК), №4, 2015, с.177-182.
13. В.В. Золотарёв, Г.В. Овечкин. Способ работы символьного порогового элемента в символьном мажоритарном декодере. Патент РФ №2573741.
14. Кларк Дж., Кэйн Дж. Кодирование с исправлением ошибок в системах цифровой связи. — М.: Радио и связь, 1987. — 391 с.

УДК 356-161.4 ГРНТИ 06.01.05

КРИПТОВАЛЮТЫ В СОВРЕМЕННОМ МИРЕ

Н.Р. Балабанов

*Рязанский государственный радиотехнический университет,
Россия, Рязань, e-mail: balabanov98.16@mail.ru*

Аннотация. Криптовалюты – всё более развивающаяся часть экономики. По сравнению с обычными валютами криптовалюты никак не контролируются и не обеспечиваются ни одним государством, более того, по законам многих стран, они даже не имеют ценности, а также их невозможно отследить.

Ключевые слова. Криптовалюта, валюта

CRYPTO-CURRENCIES IN THE MODERN WORLD

N.R. Balabanov

Ryazan State Radio Engineering Institute,
Russia, Ryazan, e-mail: balabanov98.16@mail.ru

Abstract. Crypto-currencies are an increasingly developing part of the economy. Compared to conventional currencies, crypto-currencies are not controlled and provided by any state, moreover, according to the laws of many countries, they do not even have value, and they can't be tracked.

Keywords. Crypto currency, currency

Основной принцип криптовалют – количество единиц каждой такой валюты, например известной валюты «Биткоин», строго ограничено, что исключает возможность инфляции. То есть, в отличие от обычных валют, которые, в теории, можно печатать бесконечно много, максимальное количество «монет» криптовалюты «Биткоин» будет достигнуто примерно в 2140-м году. Количество это составит примерно 20 999 999.9769 монет. Даже если увеличить точность, то число никогда не превысит 21 миллион.

Ценность криптовалют не обеспечивается золотом, нефтью или государством. Ценность криптовалют в их свободе. Без поддержки и контроля со стороны банков и государств, любая криптовалюта является свободным и независимым платёжным средством. Криптовалюты на данный момент даже не облагаются налогом. Фактически, ценность криптовалюты обеспечена договором людей, её использующих – люди готовы покупать и продавать криптовалюту по такой цене, по которой криптовалюту покупают и продают другие люди.

Основной принцип добычи новых единиц криптовалют – вычислительная мощность компьютеров. Новые монеты выдаются системой через определённое время, которое экспоненциально растёт с каждой выданной монетой. Вычислительная мощность также требуется для поддержания всей системы – каждая транзакция должна быть зашифрована и записана, по крайней мере, в 51% всех существующих единиц данной криптовалюты. И поэтому даже после добычи всех единиц какой-либо криптовалют, майнинг – добыча криптовалюты – этой валюты остановлен не будет – вычислительная мощность будет тратиться на перезапись каждой «монеты» в сети и поддержание этой самой сети. Таким образом создаётся закрытая и самодостаточная система.

Стоит отметить, что криптовалюты – не просто абстрактные «цифры» на счетах, – они являются реальными файлами, которые можно хранить на жёстком диске, флешке или в облаке. На волне успеха криптовалюты «Биткоин» стали появляться десятки новых криптовалют, которые основаны на похожих принципах.

В данной работе представлена идея ещё одного типа криптовалют.

Каждая единица криптовалюты требует мощности для создания и поддержания всей системы. Криптовалюты нельзя потратить и нельзя создать слишком много.

Но что, если создать криптовалюту, которая будет являться единицей вычислительной мощности? Сейчас сложно представить такое, но, с развитием квантовых компьютеров, криптовалюты могут потерять свою ценность – слишком прост будет процесс создания новых «монет».

Однако эти же самые квантовые компьютеры можно использовать как основание для нового типа криптовалют. Сервер, основанный на квантовых компьютерах, может являться аналогом «золотого запаса» для криптовалюты. Квантовые компьютеры могут обеспечить скорость вычислений, которая в сотни тысяч раз превосходит скорости современных компьютеров, а значит, эта скорость быстро станет востребованной. Даже для майнинга других криптовалют.

На основе суперкомпьютеров можно создать криптовалюту, каждая единица которой будет равна части вычислительной мощности этого компьютера. И каждую единицу можно будет обменять на временное использование этой мощности. Общее количество такой валюты будет ограничиваться предельной мощности квантового суперкомпьютера, часть такой

валюты всегда будет в процессе создания, а часть в процессе израсходования. Это создаст бесконечный круговорот этой валюты.

Применения такой валюты могут быть самые разные. Кроме функции обмена и торговли, она также может быть потрачена для создания других криптовалют, для быстрых расчётов, быстрого взлома различных шифров, для проведения банковских операций и так далее.

Подобная валюта может стать основой экономики в будущем, когда информация станет более дорогой, чем ресурсы. А обработка информации будет основой экономик многих развитых стран мира.

В заключение стоит сказать, что, однако, никто не знает, будут криптовалюты стабильны в будущем, или же они сохраняют тенденцию к скачкообразным изменениям цены, которые делают их слишком ненадёжны для торговли на мировых фондовых биржах.

Однако же, хотелось бы добавить, что отсутствие контроля, обвиняемое многими в нестабильности и ненадёжности, также является и гарантом иного рода стабильности криптовалют.

Ещё одним плюсом отсутствия контроля можно считать количество криптовалют – кроме самой известной криптовалюты «Биткоин» существуют десятки других, к примеру «Эфириум» (Ethereum). Причём многие криптовалюты не являются простыми клонами «Биткоин», а вносят в технологию «Блокчэйн» - на которой основаны все криптовалюты – что-то новое.

Заккрытие или запрет одной криптовалюты лишь посредственно повлияет на курсы других криптовалют. Каждая из них полностью независима от других.

Каждая обычная валюта, так или иначе, зависит от политики – таким образом крупные мировые державы могут буквально обесценить валюту других стран. Аналогично, сами страны могут непродуманными действиями обрушить цену своей валюты – как пример можно привести Зимбабве, где гиперинфляция достигала миллионов процентов в год.

Да, из-за отсутствия контроля банков и правительств никто не вернёт деньги, если «Биткоин» или другая криптовалюта рухнет. Но и обрушить криптовалюты сложнее, ведь отсутствуют политические и экономические рычаги давления. Также и отсутствует человеческий фактор, который может привести к финансовым кризисам и дефолтам.

Стоит отметить некоторую спорность этого заявления – ограничение торговли криптовалютами в США и Китае за последний год привели к нескольким крупным падениям, при которых, к примеру, «Биткоин» падал в стоимости почти в два раза. Однако каждый раз «Биткоин» возвращал позиции и, несмотря на всё, сохранял тенденцию к росту. Аналогично, поддержание валют государством и банками – не панацея. Многие обвалы валют в мировой истории приводили к большим потерям денег у обычных людей, которые никто не возмещал.

Сейчас можно сказать лишь то, что криптовалюты – это кардинально новый шаг в развитии экономики, который нельзя отрицать или игнорировать. В конце концов, когда в ходу появились бумажные деньги, то люди сначала тоже не верили в их надёжность.

Забавно отметить тот факт, что ценность обеспеченной валюты и криптовалюты имеют весьма интересную общую черту – несмотря на то, что обеспеченная валюта является эквивалентом какого-то объёма, к примеру, золота в запасе страны или банка, этим эквивалентом она является лишь на бумаге – иными словами, до тех пор, пока вы не потребуете обменять валюту на запас того, чем она обеспечена. Это негласное соглашение делает обычную валюту и криптовалюту очень похожими друг на друга. Многие критики забывают об этом.

Библиографический список

1. Манькова А.О. Криптовалюта: биткоин и его развитие в современной экономике // Молодежный научный вестник. 2016. С. 80 – 86. [Электронный ресурс]. URL: www.mnvnauka.ru/2016/02/Mankova.pdf (дата обращения: 27.12.2016).
2. Дурдыева Д.А., Трапизонян А.А. Состояние криптовалютного рынка и перспективы развития биткоин // Инновационная наука. № 1-1. 2017. С. 43 – 47.

Работа выполнена под руководством профессора кафедры Вычислительной техники ФГБОУ ВО «Рязанский государственный радиотехнический университет» Л.А. Демидовой.

СЕКЦИЯ «ЭВМ И СИСТЕМЫ»

УДК 007.52; ГРНТИ 55.30.31

ОСНОВНЫЕ ПОДХОДЫ К РЕШЕНИЮ ЗАДАЧИ SLAM В РОБОТОТЕХНИКЕ

В.А. Саблина, А.Д. Сергеева

Рязанский государственный радиотехнический университет,

Россия, Рязань, sablina.v.a@evm.rsreu.ru

Аннотация. В данной работе проведен сравнительный анализ основных подходов к решению задачи одновременной локализации и картографирования (SLAM), акцентируется внимание на сильных и слабых сторонах каждого из алгоритмов.

Ключевые слова. SLAM, EKF-SLAM, FastSLAM, DP-SLAM, LSD-SLAM, Gamma-SLAM, фильтр Калмана.

THE MAIN APPROACHES TO SOLVING THE PROBLEM OF SLAM IN ROBOTICS

V.A. Sablina, A.D. Sergeeva

Ryazan State Radio Engineering University,

Russia, Ryazan, sablina.v.a@evm.rsreu.ru

Abstract. This paper provides the comparative analysis of some methods solving the problems of simultaneous localization and mapping (SLAM), focuses on the strengths and weaknesses of each of the algorithms.

Keywords. SLAM, EKF-SLAM, FastSLAM, DP-SLAM, LSD-SLAM, Gamma-SLAM, Kalman filter.

Введение

Задача, основанная на одновременной локализации и картографировании, в настоящее время известна как SLAM (Simultaneous Localization and Mapping). Ей уделяется огромное внимание в робототехнике. Исходными данными задачи являются: некоторая среда (помещение, водная или воздушная среда) и робот с датчиками системы технического зрения (дальномеры, одометры, видео камеры).

Среды функционирования классифицируются следующим образом: с ярко выраженными ориентирами и с отсутствием ярко выраженных ориентиров. Если изучаемая среда не содержит ярко выраженных ориентиров, тогда ее представляют в виде массива, состоящего из нулей и единиц. Нулем обозначают отсутствие ориентира, единицей – наличие. Для хранения карты ориентиров используют специальную картографическую базу данных, содержащую информацию о положении ориентиров, их уникальных свойствах и взаимосвязях.

Основными подходами к решению задачи SLAM являются: EKF-SLAM, FastSLAM, DP-SLAM, LSD-SLAM, Gamma-SLAM. Рассмотрим каждый из них подробнее.

Метод EKF-SLAM

Класс алгоритмов EKF-SLAM (от англ. Extended Kalman Filter SLAM) использует расширенный фильтр Калмана для одновременной локализации и картографирования. Данные алгоритмы основываются на максимальном правдоподобии с целью объединения данных. Ключевым ограничением подходов, основанных на EKF, является их вычислительная сложность. Обновление датчика требует квадратичного времени вычислений, зависящего от количества объектов в окружающей среде, что влечет за собой ограничение (до нескольких сотен) на количество ориентиров, которые могут быть обработаны с помощью данного подхода, в то время как модель окружающей среды, как правило, содержит миллионы ориентиров, что признано существенным недостатком алгоритма [1].

Метод FastSLAM

Основой алгоритма FastSLAM является фильтр частиц и применяемая в нем Байесовская сеть (или динамическая сеть Байеса) (рис. 1), в которой позиции робота, обозначаемые как $S_1, S_2, \dots, S_n$, изменяются с течением времени как функции управления роботом, обозначаемые $u_1, u_2, \dots, u_n$. $Z_1, Z_2, \dots, Z_n$ обозначают функцию положения ориентира и робота во времени. Наблюдаемые роботом ориентиры в каждый момент времени обозначаются как θ_1, θ_2 из множества ориентиров $\{\theta_1, \theta_2\}$. Серая область иллюстрирует условие независимости положения робота и ориентиров.

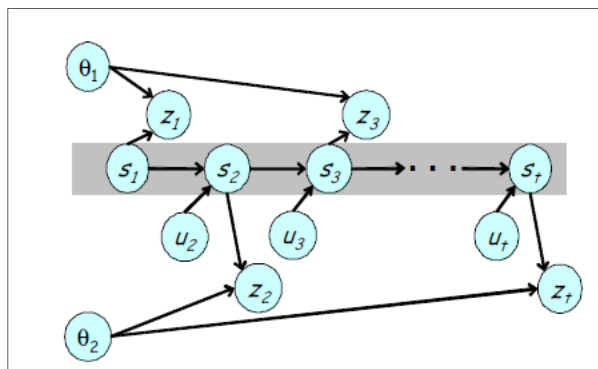


Рис. 1. Динамическая сеть Байеса

В FastSLAM одна большая карта рассматривается как совокупность локальных подкарт, что позволяет убрать зависимость ориентиров друг от друга и таким образом значительно сократить время пересчета оценки состояния системы. На основе данного замечания, можно сделать вывод, что проблемы локализации робота и сбора оценок ориентиров, обусловлены оценкой положения робота.

В данном методе используется модифицированный фильтр частиц для оценки последующего пути робота. Каждая частица, обладающая фильтром Калмана, оценивает положения ориентиров, основываясь на рассчитанном пути. Полученный алгоритм является разновидностью фильтра частиц Рао-Блэкуэллза (Rao-Blackwellized) [2]. Простая реализация этой идеи приводит к алгоритмам, которые требуют $O(MK)$ времени, где M – число частиц в фильтре частиц, а K – число ориентиров, при переходе к древовидной структуре данных уменьшается время работы алгоритма до $O(M \ln K)$, что приводит к значительному увеличению скорости выполнения алгоритма по сравнению с классом алгоритмов EKF-SLAM.

Экспериментальные результаты с использованием физического робота и робота симулятора иллюстрируют, что FastSLAM может обрабатывать на порядок больше ориентиров, чем другие имеющиеся подходы. В некоторых ситуациях доступное количество ориентиров K приводит к уменьшению количества частиц M , необходимых для получения точной карты, в то время как в других алгоритмах для точного отображения карты требуется очень большое количество частиц. Таким образом, алгоритм FastSLAM показывает хорошее масштабирование при большом количестве ориентиров и небольшом количестве частиц. С увеличением количества ориентиров K постепенно уменьшается количество ошибок на карте и в определении положения робота (рис. 2, а). Данный процесс, в свою очередь, связан с тем, что чем больше количество частиц, тем больше ошибок в определении положения робота (рис. 2, б).

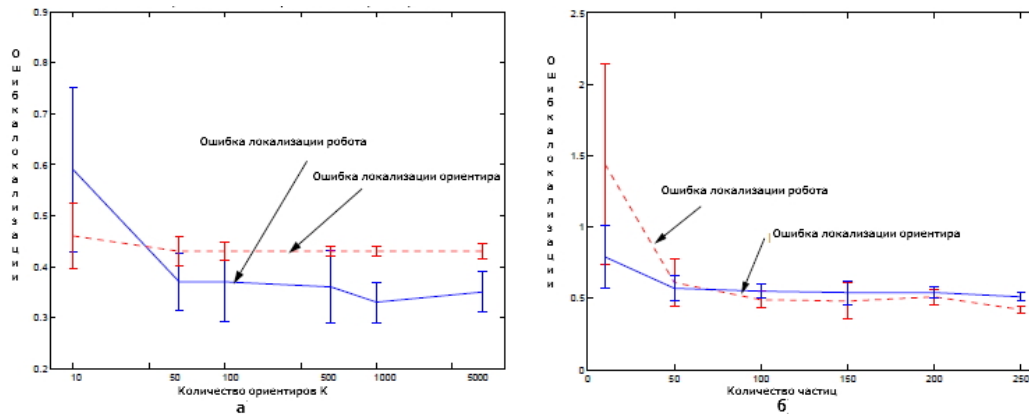


Рис. 2. Точность алгоритма FastSLAM:

а – с увеличением количества ориентиров; б – с увеличением количества частиц

Недостатком алгоритма является потенциальное падение точности, связанное с игнорированием корреляции ошибок, связанных с оценкой положений ориентиров [1].

Метод DP-SLAM

Подход DP-SLAM к решению задач SLAM реализует простой фильтр частиц и локализацию робота. Данный метод основан на применении технологии, называемой отображением распределенных частиц – distributed particle mapping (DP-Mapping) [3]. Метод DP-SLAM применяет фильтр частиц для поддержания равномерного распределения вероятностей по картам позиций робота. Карта представляется в виде сетки с заполнением ячеек, занятых препятствиями. На рис. 3 показано происхождение частиц и обновление карты, листья дерева указывают на карты. Каждая красная точка в древе предков – позиция робота, а черные линии вокруг красной точки – новые наблюдения, связанные с текущей позицией робота. Серые линии показывают фрагмент карты, унаследованный от предыдущей частицы.

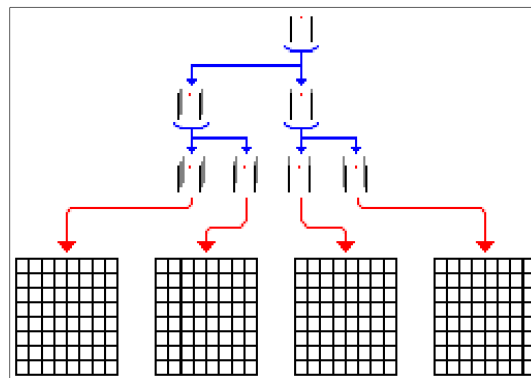


Рис. 3. Прохождение частиц и обновление карты

С целью уменьшения памяти и времени, затрачиваемых на хранение и повторное копирование разделов карты при каждой новой выборке частицы, используется единая сетка наблюдений, которая хранит дерево наблюдений в каждом квадрате. Как показано на рис. 4, каждая частица вставляет свои наблюдения в глобальную сеть. Они хранятся в виде сбалансированного дерева, проиндексированного по уникальному идентификатору, присвоенному каждой частице.

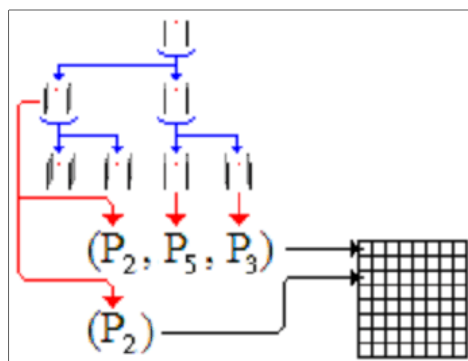


Рис. 4. Сбалансированное дерево наблюдений

Таким образом, при применении подхода DP-SLAM следует уделить особое внимание выбору структуры данных во избежание больших затрат времени и памяти при работе алгоритма. Данное утверждение признано недостатком метода DP-SLAM [4].

Метод LSD-SLAM

Метод LSD-SLAM (от англ. Large-Scale Direct Monocular SLAM) основан на непосредственном (без использования признаков) определении местоположения и картографировании. В основе данного подхода используются системы с одиночной камерой (монокулярный SLAM) или стереосистемы камер, в пределах эффективного рабочего диапазона которых обеспечивается точная глубина получаемого изображения, устраняется неоднозначность масштаба и трудности с ухудшением перемещения вдоль линии видимости, что присуще монокулярному SLAM, который использует только временное стерео. С другой стороны глубина изображения может быть оценена с помощью переменного исходного направления, которое соответствует движению между кадрами.

Таким образом, если объединить статическое и временное стерео становится доступным несколько базовых направлений, которые позволяют оценить глубину вдоль горизонтальных кромок, чтобы построить карту глубины. Данный подход при систематической оценке двух наборов тестовых данных из робототехнических приложений демонстрирует хорошую производительность.

Общее описание алгоритма подхода Stereo LSD-SLAM проиллюстрировано на рис. 5 [5].

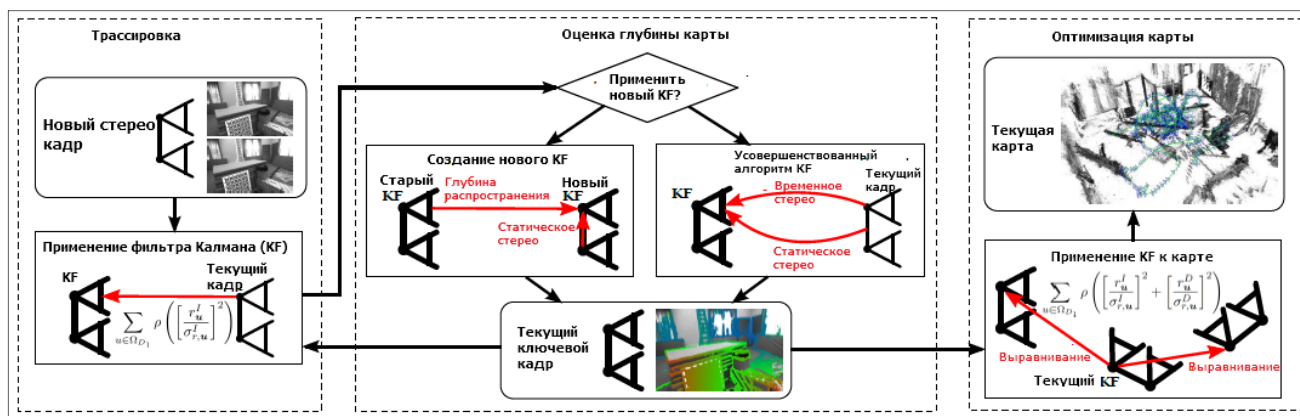


Рис. 5. Представление системы Stereo LSD-SLAM

Метод Gamma-SLAM

Метод Gamma-SLAM решает проблему оптимального преобразования карты SLAM в геодезические координаты, основанные на шуме GPS-сигнала. В результате получается онлайн-версия этого процесса преобразования, который может быть использован дополнительно при выполнении оценки глобального позиционирования робота, которая является гораздо более точной, чем показания GPS.

В данном методе, как и в рассмотренных ранее, для получения информации о локализации и картографировании, используется фильтр частиц Rao-Blackwellized: распределение положения оценивается с помощью фильтра частиц, а каждая частица имеет свою собственную карту, которая получена путем точной фильтрации, обусловленной положением частицы. Визуальная одометрия обеспечивает хорошее представление распределений для положения частицы. В аналитическом (точном) фильтре для карты обновляется необходимая и достаточная статистика гамма – распределения по точности высот в каждой ячейке сетки.

Данная система использует плотное стерео зрение и карты, содержащие постериорное (posterior) распределение дисперсии высот в каждой ячейке. В отличие от заселенных сетей (в которых предполагается, что каждая ячейка в окружающей среде либо занята, либо свободна) и карт высот (перепадов) (в которых предполагается, что каждая ячейка в окружающей среде является плоской поверхностью), данные карты основаны на предположении, что точки, наблюдаемые из каждой ячейки, взяты из Гауссова распределения высоты. Вместо того чтобы оценивать среднюю высоту ячейки, метод оценивает дисперсию высот. Карты отклонений не ограничиваются плоскими поверхностями, которые имеют решающее значение для неструктурированных наружных сред. Рассматриваемый метод отображения поддерживает постериорное (posterior) распределение по дисперсии высот точек в каждой ячейке сетки. Для каждой из ячеек сетки постериорное (posterior) распределение по точности (обратное отклонение) высот представляет собой гамма-распределение (поэтому алгоритм назван Gamma-SLAM). Каждая ячейка карты содержит два скалярных значения, которые обладают достаточной статистикой гамма-распределения.

Роботизированное транспортное средство, которое используется в данном методе, имеет GPS-приемник. Ошибка в сигнале GPS оказывается в экспериментах достаточно велика по сравнению с точностью получаемых SLAM-карт. SLAM-карты обеспечивают точную информацию относительно местоположения, которая однако не совсем согласована с Землей. При объединении данных получаемых с помощью метода Gamma-SLAM с неточными данными с GPS-приемника были получены данные о положении робота в координатах гораздо более точных, чем при применении только GPS-приемника (рис. 6). Так при перемещении робота на расстояние 204 метра ошибка при использовании визуальной одометрии составляет 1,12 метров, а при использовании Gamma-SLAM – 0,21 метра.

На рис. 6, а путь робота по значениям GPS показан красным цветом. Можно обратить внимание на ошибки GPS, такие как случайные внезапные прыжки в значениях GPS от одного момента времени к другому. Красная точка показывает значение GPS в начале, а пунктирный красный круг (с центром в красной точке) показывает горизонтальную погрешность установки, данную устройством GPS. Синяя линия – это путь робота, полученный на основе метода Gamma-SLAM, перемещения и повороты выполнены с целью минимизации ошибки GPS. Синяя точка указывает положение робота в начале эксперимента. На рис. 6, б представлено сравнение горизонтальной погрешности установки, данное устройством GPS (красная линия) и ошибки SLAM. Решение SLAM укладывается в пределы погрешности данных устройства GPS [6].

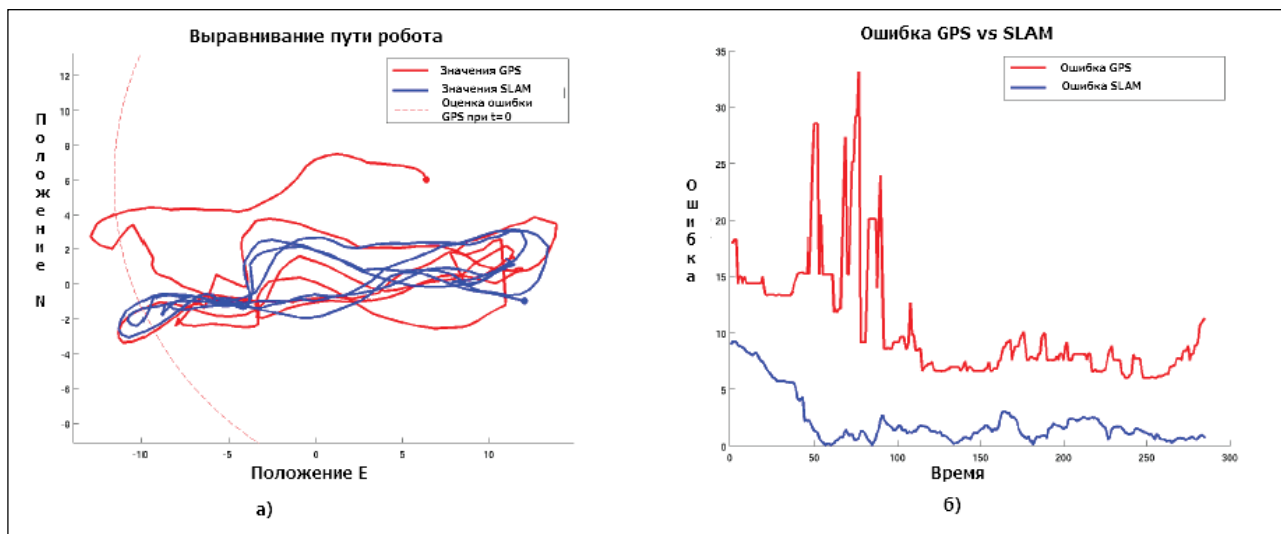


Рис. 6. Сравнение результатов Gamma-SLAM и GPS: а – Точность определения положения робота по методу Gamma-SLAM и GPS; б – сравнение оценок ошибок в определении по GPS и по методу Gamma-SLAM

Выводы

В статье рассмотрены особенности технологии SLAM, а также основные варианты ее реализации: EKF-SLAM, FastSLAM, DP-SLAM, LSD-SLAM, Gamma-SLAM. Технология SLAM продолжает развиваться и находить применение в различных системах технического зрения. Такие системы могут быть основаны на комбинациях различных датчиков. Работы в данном направлении ведутся в ведущих университетах и научных центрах мира. Таким образом, представляется перспективным дальнейшее исследование возможностей применения технологии SLAM на практике.

Библиографический список

1. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," In Proc. Of the Nat. Conf. on Artificial Intelligence, 2002, pp. 593-598.
2. A. Doucet, N. de Freitas, K. Murphy, and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," In Proc. of the Sixteen conference on Uncertainty in artificial intelligence, 2000, pp. 176-183.
3. A. Eliazar, R. Parr, "DP-SLAM 2.0" [Электронный ресурс]. URL: <http://www.cs.duke.edu/~parr/dpslam2.pdf> (дата обращения: 15.10.2017).
4. A. Eliazar, R. Parr, "How DP-SLAM Works" [Электронный ресурс]. URL: https://users.cs.duke.edu/~parr/dpslam/#How_DP-SLAM_Works (дата обращения: 30.09.2017).
5. J. Engel, J. Stückler, and D. Cremers, "Large-Scale Direct SLAM with Stereo Cameras," In International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 1935-1942.
6. T. Marks, A. Howard, M. Bajracharya, G.W. Cottrell, and L. Matthies, "Gamma-SLAM: Using Stereo Vision and Variance Grid Maps for SLAM in Unstructured Environments," IEEE International Conference on Robotics and Automation (ICRA), 2008, pp. 3717-24.

УДК 621.324; ГРНТИ 50.09.47

РАЗРАБОТКА УСТРОЙСТВА СИНХРОНИЗАЦИИ ПРОЦЕССОВ ДЛЯ МНОГОПРОЦЕССОРНЫХ СИСТЕМ

Р.А. Бикташев, А.А. Петкилев, К.А. Балаев

Пензенский государственный университет,

Россия, Пенза, cnit@pnzgu.ru

Аннотация. Рассматриваются теоретические и практические вопросы разработки аппаратно - ориентированных устройств управления процессами многопроцессорных систем с использованием логики недетерминированных автоматов. Приведен пример синтеза планировщика с глобальной очередью задач, основанного на алгоритме синхронизации «рандеву».

Ключевые слова. Многопроцессорные системы, ядро операционной системы, управление ресурсами, синхронизация процессов.

DEVELOPMENT OF A PROCESS SYNCHRONIZATION DEVICE FOR MULTIPROCESSOR SYSTEMS

R.A. Biktashev, A.A. Petkilev, K.A. Balaev

Penza State University,

Russia, Penza, cnit@pnzgu.ru

Abstract. The theoretical and practical issues of the development of hardware - oriented devices for controlling the processes of multiprocessor systems using the logic of nondeterministic automata are considered. An example of a scheduler synthesis with a global queue of tasks based on the synchronization algorithm "rendezvous" is given.

Keywords. Multiprocessor systems, operating system kernel, resource management, process synchronization.

В вычислительных системах синхронизация взаимодействующих процессов является одной из функций подсистемы управления ресурсами. Она заключается в том, что запрашивающая задача анализирует глобальную блокирующую переменную, закрепленную за общим ресурсом. Операция чтения блокирующей переменной, её проверки и установки, а также записи в ячейку производится единой (неделимой) командой. В процессорах время выполнения команды «читать – проверить – установить – записать» по сравнению со временем выполнения обычных команд увеличивается в два - три раза, что приводит к простоям конвейера команд и уменьшению реакции системы прерывания.

В многопроцессорной системе (МПС) блокирующая переменная хранится в ячейке общей оперативной памяти и доступна любым задачам (процессам, потокам), в том числе и тем, которые развиваются в разных процессорных узлах. Для доступа к ней (в случае кэш – промаха) процессор должен захватить межпроцессорную шину, выполнить операцию чтения ячейки памяти, в которой хранится блокирующая переменная, провести её анализ, изменить состояние на противоположное (если общий ресурс не занят), и записать новое значение в ту же ячейку памяти.

Применение неделимой операции в многопроцессорной системе вызывает также необходимость блокировки межпроцессорной шины, функционирующей в режиме расщепления транзакций. Этим исключается возможность одновременного доступа общего ресурса со стороны нескольких процессов, выполняющихся в разных процессорах, поскольку в промежутке невыполненной транзакции шину может захватить другой процессор и обеспечить вход его процесса в свою критическую секцию. Для решения проблемы работы с блокирующими переменными в многопроцессорных конфигурациях большинству современных процессоров предоставляют аппаратные примитивы взаимного исключения: средства, позволяющие процессору монополюжно захватить шину и выполнить несколько операций над памятью.

Известно, что все современные процессоры содержат в своем составе кэш-память. Если в процедуре синхронизации участвует нескольких процессорных узлов, то возможно необоснованное увеличение трафика на межпроцессорной шине из-за частого перемещения

строки кэш, содержащей слово блокировки, которая необходима процессам, конкурирующим за доступ к общему ресурсу [1].

Последние два обстоятельства приводят к снижению реальной пропускной способности межпроцессорной шины и, в конечном счете, к потерям производительности всей вычислительной системы.

Все вышесказанное относится к синхронизации процессов в пользовательском пространстве, которая, как известно, часто приводит к фатальным ошибкам программирования [1]. В связи с этим зачастую синхронизацию процессов реализуют в ядре операционной системы. Однако для этого требуется выполнить системный вызов для входа процесса в ядро. Затраты времени на такой вход составляют несколько десятков или даже сотен процессорных тактов, поскольку он реализуется через программное прерывание. Накладные расходы этого метода приводят к дополнительным потерям производительности МПС.

Исходя из вышесказанного поставлена задача разработки с низкой латентностью устройства управления ресурсами, входящего в состав аппаратного ядра операционной системы [2]. Устройство должно обеспечивать синхронизацию n параллельных процессов для бесконфликтного доступа к общим (разделяемым) ресурсам без привлечения процессоров к выполнению данной функции, что позволит дополнительно уменьшить накладные расходы.

При разработке устройства применен формальный метод, базирующийся на логике событийных недетерминированных конечных автоматов (СНДА), позволяющий описывать все реализуемые события в стандартной форме в виде систем канонических уравнений (СКУ) [3]. Результатом формализации является система канонических уравнений автомата Мура. Для любого i -го процесса вход в критическую секцию обозначается событием S_k^i и описывается следующим уравнением [4]:

$$S_k^i = S_{vp}^i S_{vz}^i S_{pr}^i \vee S_k^i \bar{S}_p^i, \quad i = \overline{1, n},$$

где событие S_{pr}^i означает приоритет i -го запроса, а S_{vp}^i - событие, обозначающее принятие i -й заявки на обслуживание общим ресурсом. Оно описывается уравнением:

$$S_{vp}^i = (S_z^i \vee S_{vp}^i) \bar{S}_k^i,$$

где S_z^i событие, которое обозначает запрос процесса на обслуживание общим ресурсом.

S_{vz}^i - событие взаимного исключения процессов при обращении к общему ресурсу, которое описывается уравнением:

$$S_{vz}^i = \bigwedge_{(\forall \alpha)[\alpha \neq i]} \bar{S}_k^\alpha, \quad i = \overline{1, n}.$$

Событие S_{pr}^i для циклической дисциплины обслуживания с относительным приоритетом представляется следующим выражением:

Дополнительные события, представленные на графе (рис.1), определяются следующими образом:

$$\begin{aligned} S_z^i(t+1) &= x_z^i \vee S_z^i \overline{S_{vp}^i}, & S_r^i(t+1) &= (S_{vp}^1 \vee S_r^i) \overline{S_p^i}, \\ S_m(t+1) &= S_k^1 S_{vp}^1 \vee S_k^2 S_{vp}^2 \vee \dots \vee S_k^n S_{vp}^n, & S_p^i(t+1) &= S_{nk} S_k^i, \\ S_{nk}(t+1) &= S_m, & S_{pk}^i(t+1) &= S_p^i \vee S_{pk}^i S_{np}^i S_T, \\ S_j^i(t+1) &= S_r^i S_p^i. \end{aligned}$$

где x_z^i - сигнал обращения к общему ресурсу i -го процесса;

S_m - сокращенное обозначение события, обеспечивающего начало процедуры реализации обращения к разделяемым данным;

S_{nk} - сокращенное обозначение события, являющегося заключительным в процедуре обращения к разделяемым данным;

S_r^i - сокращенное обозначение события, символизирующего ожидание условий выхода i -го процесса из критического участка;

S_j^i - сокращенное обозначение события, инициирующего продолжение работы i -го процесса после выхода из критического интервала.

Объединяющая вершина графа $J(\vee)$ реализует функцию дизъюнкции конъюнкций переменных S_k^i и S_{vp}^i .

Схема устройства синхронизации (УС) представлена в виде трех блоков (рис. 2а). Блок 1 регистрирует запросы к общему ресурсу и фиксирует вход i -го процесса в критический интервал. На рис. 2а введены следующие обозначения: S_z^i - событие (сигнал) запроса на доступ к общему ресурсу от i -го процесса, выполняющегося в одном из процессоров; S_k^i - событие (сигнал) разрешения входа в критическую секцию (КС) i -го процесса; S_p^i - событие (сигнал) завершения работы в КС i -го процесса; S_j^i - событие (сигнал) разрешения для продолжения выполнения i -го процесса, причем i принимает значения от 1 до n ($i = \overline{1, n}$). Блоки 2 и 3 реализуют функции взаимоисключения и выделения приоритетного запроса.

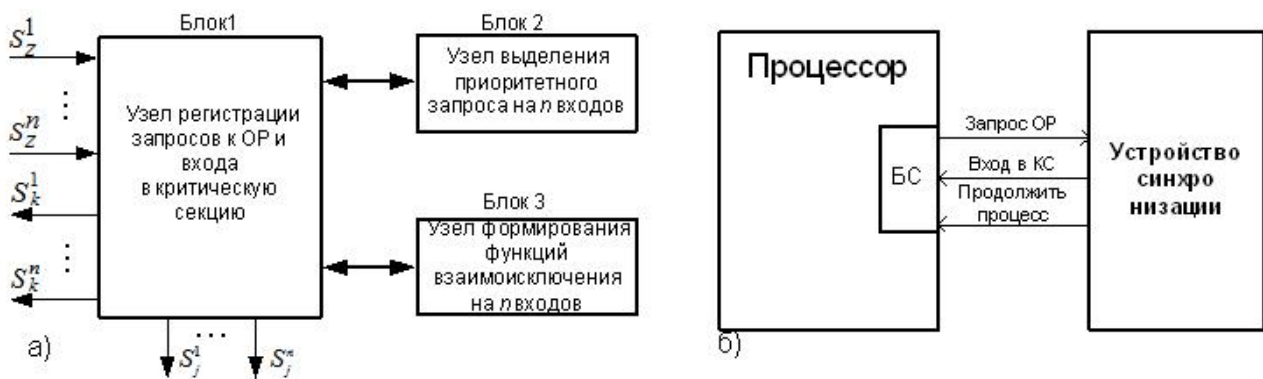


Рис. 2. Модель устройства синхронизации на n процессов (а) и интерфейс взаимодействия устройства синхронизации с процессором (б)

На рисунке 2б показана схема интерфейса «процессор – УС». Процессы, выполняющиеся в процессорах, формируют запросы на доступ к общему ресурсу (Запрос ОР - S_z^i), которые через специальный блок сопряжения (БС) поступают в УС. Устройство синхронизации, при незанятости ресурса, уведомляет о принятии запроса и разрешает процессу вхождение в критическую секцию (Вход в КС - S_k^i). После выполнения необходимых действий над общим ресурсом процесс выходит из критической секции, после чего УС разрешает продолжение выполнения процесса (по сигналу «Продолжить процесс» - S_j^i).

Представленная выше система канонических уравнений формально описывает автомат Мура, который легко трансформируется в VHDL модель. Эта Модель в виде четырехканального устройства (на 4 запроса) была реализована на языке VHDL и проанализирована в различных режимах (рис. 3).

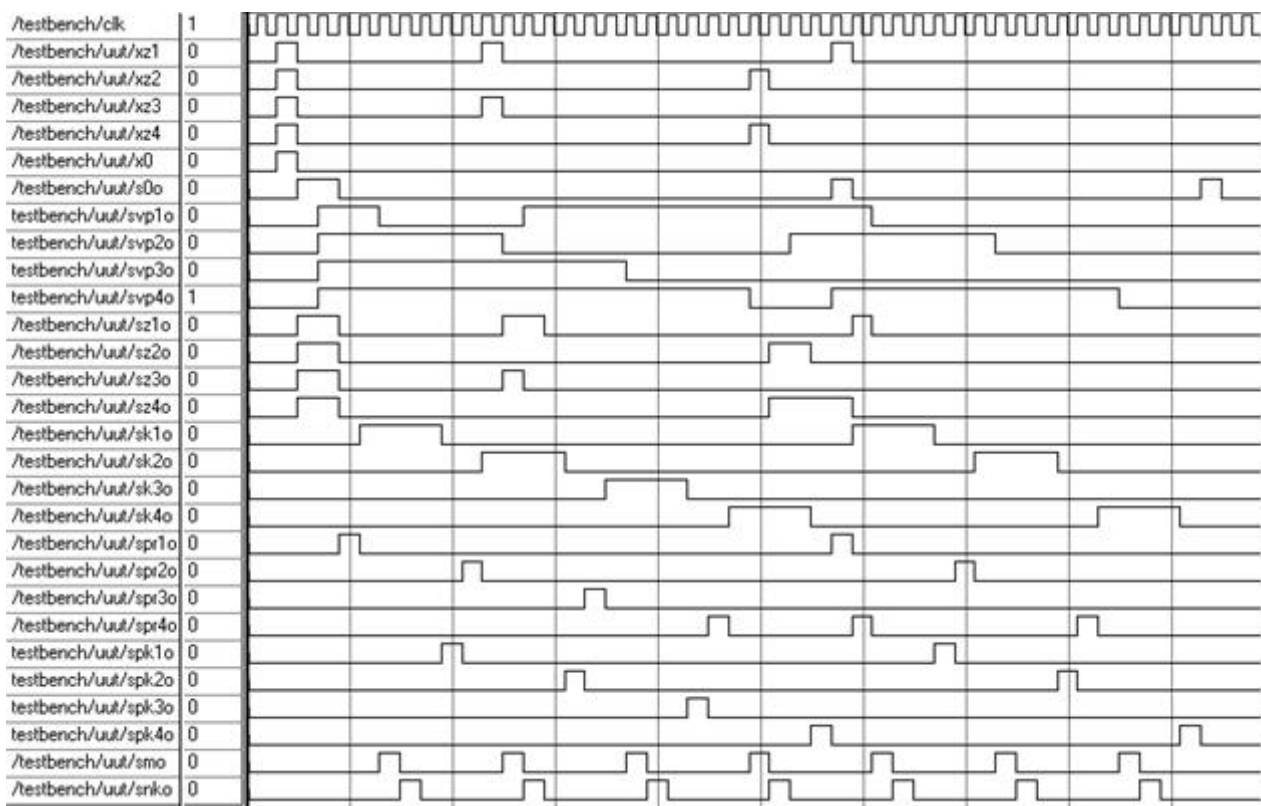


Рис.3. Результаты моделирования устройства синхронизации

Результаты моделирования показывают, что латентность запроса (задержка от момента поступления запроса до момента начала работы процесса в критической секции в условиях бесконфликтности) составляет всего пять процессорных тактов, что значительно меньше, чем при традиционной программной реализации подобной ИРС-функции в разделяемой памяти, которая по оценкам зарубежных исследователей составляет несколько сотен тактов. А латентность подобной реализации критической секции в ядре ОС составляет от нескольких сотен до нескольких тысяч процессорных тактов в зависимости от типа процессора [5]. Поскольку процессор освобождается от выполнения части программы, которая обеспечивает вход в критическую секцию, то сэкономленные такты могут быть использованы для выполнения других функций (например, других приложений), а показатели производительности компьютерной системы в зависимости от частоты обращения процессов к общим ресурсам по предварительным оценкам могут возрасти на 10-25%.

Разработанное устройство синхронизации может быть включено в аппаратное ядро многопроцессорной операционной системы [2] и использовано в системах жесткого реального времени, где повышены требования к времени реализации приложений.

Библиографический список

1. Таненбаум Э., Бос Х. *Современные операционные системы*. 4-е изд. — СПб.: Питер, 2015. — 1120 с.
2. Бикташев Р.А. [Архитектура аппаратного ядра многопроцессорной операционной системы](#) // В сборнике: [Новые информационные технологии и системы](#). НИТиС-2015: Пенза, 2015. С. 86-90.
3. Волчихин В.И., Вашкевич Н.П., Бикташев Р.А. Модели событийных недетерминированных автоматов представления алгоритмов управления взаимодействующими процессами в многопроцессорных вычислительных системах на основе использования механизма монитора// Известия высших учебных заведений. Поволжский регион. Технические науки. 2013, № 2, с.5-14.
4. Вашкевич Н.П. Недетерминированные автоматы и их использование для реализации систем параллельной обработки информации / Н.П.Вашкевич, Р.А. Бикташев. — Пенза: Изд-во ПГУ, 2016. - 394с.
5. Stoif, C., Schoeberl, M., Liccardi, B., & Haase, J. Hardware Synchronization for Embedded Multi-Core Processors. In Proceedings of the 2011 IEEE International Symposium on Circuits and Systems ISCAS 2011, pp. 2557-2560.

УДК 004.93; ГРНТИ 28.23.15

РАЗРАБОТКА БЕЗЭТАЛОННОГО ИНТЕГРАЛЬНО-МУЛЬТИПЛИКАТИВНОГО ПОКАЗАТЕЛЯ КАЧЕСТВА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

А.С. Сычев

*Рязанский государственный радиотехнический университет,
Рязань, alekseisichev2203@gmail.com*

Аннотация. Цель работы – разработка безэталонного интегрально-мультипликативного нормированного показателя качества цифровых полутоновых изображений для оценки эффективности работы алгоритмов улучшения видения и выбора каналов при комплексировании информации от разноспектральных сенсоров. Введенный показатель оперирует оценками средней яркости, среднеквадратического отклонения яркости высокочастотных компонент и шума, а также средними значениями локальных контрастов анализируемого кадра и его низкочастотной составляющей. Результаты полунатурного эксперимента показали, что численные значения интегрально-мультипликативного критерия в отличие от известного интегрального показателя при низких отношениях сигнал-шум лучше согласуются с субъективным восприятием качества изображения.

Ключевые слова. Безэталонный критерий качества, интегральный показатель качества, яркость, среднеквадратическое отклонение, гистограмма, контраст, уровни яркости, энтропия.

DESIGN OF NO-REFERENCE INTEGRATED-MULTIPLICATIVE QUALITY INDEX FOR DIGITAL IMAGES

A.S. Sychev

*Ryazan State Radioengineering University, Ryazan
alekseisichev2203@gmail.com*

Abstract. The aim of the work is the development of a no-reference integrated-multiplicative normalized quality index for digital grayscale images for evaluating the efficiency of algorithms for enhanced vision and selecting informative channels for multispectral image fusion. The introduced quality index operates with evaluations of the average brightness, standard deviation of the the high-frequency components brightness and noise, as well as the average values of the local contrasts of the analyzed frame and its low-frequency component. The results of the semi-real experiment showed that the numerical values of the integral-multiplicative criteria, in contrast to the known integral quality index, are in better agreement with the subjective perception of image quality at low signal-to-noise ratios.

Keywords. No-reference quality criteria, integral quality index, brightness, standard deviation, histogram, contrast, brightness levels, entropy.

Введение

В различных информационных системах применяется представление результатов обработки данных в виде изображения, выводимого на устройство отображения для использования наблюдателем (человеком-оператором) – визуализация. Выводимому изображению при этом желательно обеспечить такие качества, благодаря которым его восприятие было бы по возможности комфортным, усилив информативные особенности наблюдаемой сцены [1]. Под сценой, как и в [1], будем понимать все объекты, попадающие в кадр фото- и/или видео-камеры.

Качество изображения является субъективным понятием, что усложняет разработку формального подхода для его оценки. Оно зависит от ряда факторов [1]:

- освещенности и контрастности сцены,
- смаза изображения в результате движения камеры или объекта съемки,
- глубины резкости,
- способа квантования зарегистрированного сенсором сигнала,
- способа кодирования и степени сжатия при записи в файл.

Численные оценки показателей качества цифровых полутоновых изображений помимо собственно ранжирования последних по данному параметру могут применяться также для выбора алгоритмов и параметров преобразований, ориентированных на улучшение визуализации [2], а также на выбор информативных каналов в разноспектральных системах улучшенного видения [3].

Меры оценки качества цифровых изображений

Показатели качества изображений можно условно разделить на два класса [1]:

- меры сравнения с эталоном, в которых одно изображение X считается эталонным, а второе Y – преобразованным, применимые при анализе алгоритмов сжатия изображений и их искусственного искажения при имитационном моделировании, поскольку в обоих случаях имеется исходное изображение-оригинал (эталон);

- безэталонные меры, оперирующие статистиками только текущего анализируемого изображения, когда эталонное изображение отсутствует: например, в обзорных оптико-электронных системах необходимо выполнить преобразования для улучшения визуализации зафиксированного камерой изображения и сразу вывести его на устройство отображения. В этом случае количественными показателями качества изображения могут выступать средняя яркость и СКО (среднеквадратическое отклонение) яркости по полю изображения, нормированные контраст и количество информационных уровней, а также энтропия [4].

Оценка визуального качества изображения по отдельным мерам не является объективной. По этим причинам целесообразно при оценке визуального качества оперировать комплексными критериями.

Комплексные безэталонные критерии качества цифровых полутоновых изображений

Комплексные критерии качества формируют единственный (как правило, нормированный на интервале $[0, 1]$) показатель качества Q из n частных показателей α_i , $i = 1..n$, и могут представлять собой их произведение (мультипликативный показатель качества, МПК)

$$Q = w_n \prod_{i=1}^n \alpha_i, \quad (1)$$

либо весовую сумму (интегральный показатель качества, ИПК)

$$Q = \sum_{i=1}^n w_i \alpha_i,$$

где нормирующий множитель w_n приводит Q к диапазону $[0, 1]$, а для весовых коэффициентов w_i справедливо равенство $\sum_{i=1}^n w_i = 1$.

Примером МПК является эмпирический критерий качества Воробеля [5]:

$$Q_V = w_n L_Q K_Q R_Q K_{nV}, \quad (2)$$

где L_Q – оценка уровня адаптации зрительной системы;
 $K_Q = S/L_{\max}$, – полнота градаций яркости;
 S – количество уровней яркостей, для каждого из которых на изображении присутствует не менее, чем bWH количество пикселей с данной яркостью;
 b – некоторая константа;
 $R_Q = RO/L_{\max}$ – резкость изображения RO , нормированная к максимальной яркости $L_{\max}$;
 K_{nV} – контраст по Воробелю [5];
 w_n – нормирующий множитель.

Недостатком МПК Воробеля является равный вес всех входящих в него частных показателей и близость к нулю Q_V даже при величине каждого показателя качества около 0,5 (действительно, неравенство $Q_V < 0,1$ выполняется уже при $L_Q = K_Q = R_Q = K_{nV} < 0,5$).

В работе [4] проблема выбора весов частных показателей качества изображения решена предварительным их разбиением на три группы с убывающим приоритетом (математическое ожидание и СКО яркости, контраст, количество градаций яркости и энтропия) и последующим применением метода экспертных оценок и критерия Фишберна. В результате получено следующее эмпирическое выражение для ИПК:

$$Q_{\text{ИПК}} = 0,33L_n + 0,27\sigma_n + 0,2K_n + 0,13N_n + 0,07\varepsilon_n, \quad (3)$$

где L_n , σ_n , K_n , N_n и ε_n – соответственно нормированные средняя яркость, СКО, глобальный контраст, количество уровней и энтропия. Перечисленные частные нормированные показатели рассчитывают по аналогичным ненормированным величинам:

- средней яркости $\bar{L} = \sum_{i=0}^{255} i h(i)$, где $h(i)$ – значения гистограммы, соответствующие уровню яркости i ,

- СКО яркости по полю изображения $\sigma = \sum_{i=0}^{255} (i - \bar{L})^2 h(i)$, (4)

- динамическому диапазону $\Delta L = L_{\max} - L_{\min}$, где $L_{\min}$ и $L_{\max}$ – соответственно минимальная и максимальная яркость,

- количеству информационных уровней яркости

$$N_{\text{я}} = \sum_{i=0}^{255} \delta(i), \quad \delta(i) = \begin{cases} 1, & h(i) > 0, \\ 0, & h(i) = 0, \end{cases}$$

и энтропии $\varepsilon = -\sum_{i=0}^{255} \log_2[h(i)]h(i)$ по следующим формулам:

$$L_n = \begin{cases} \bar{L} / 128, & \bar{L} \leq 107, \\ (255 - \bar{L}) / 128, & \bar{L} > 147, \\ 1, & \bar{L} \in (107 \dots 147), \end{cases} \quad N_n = N_{\text{я}} / 255, \quad \varepsilon_n = \varepsilon / 8, \quad (5)$$

$$\sigma_n = \begin{cases} \sigma / 50, & \sigma \leq 50, \\ (100 - \sigma) / 50, & \sigma > 50, \end{cases} \quad K_n = \Delta L / 255. \quad (6)$$

ИПК применяется в работах [2, 4] для оценки качества результатов работы алгоритмов улучшения видения.

Модифицированный интегрально-мультипликативный показатель качества полутоновых изображений

Значения ИПК при анализе изображений с низким уровнем шума в целом совпадают с субъективными оценками их качества. Однако в случае увеличения СКО шума на изображении статистика (3) стремится к значениям 0,9...1 независимо от сюжета сцены. Это объясняется тем, что шум увеличивает все входящие в статистику (3) частные показатели. В частности, для аддитивной смеси серого фона с яркостью $L = 128$ и белого гауссовского шума (БГШ) с нулевым математическим ожиданием и СКО $\sigma = 50$, не несущих никакой информации, выполняется $Q_{\text{ИПК}} > 0,95$. Это проиллюстрировано в разделе эксперимента на рис. 1 и в табл. 1.

Вторым недостатком ИПК является то, что оценка СКО (4) является показателем резкости изображения только в том случае, если форма его гистограммы хорошо аппроксимируется гауссовской кривой.

Также к недостаткам ИПК можно отнести разрыв функционала (5) в точках $\bar{L} = 107$ и $\bar{L} = 147$ и статистическую зависимость частных показателей N_n и ε_n (с ростом количества уровней яркости будет увеличиваться и энтропия).

Автор предлагает оценивать качество полутоновых изображений по частным показателям, для которых можно компенсировать влияние шума сенсора. Для этого принимаются следующие гипотезы:

- 1) шум сенсора описывается математической моделью аддитивного БГШ с нулевым математическим ожиданием;
- 2) формат изображений – полутоновые, с глубиной цвета 8 бит.

При принятии данных гипотез вводится интегрально-мультипликативный показатель качества (ИМПК), который записывается в виде:

$$Q_{\text{ИМПК}} = L_Q [w_1 w(q) \sigma_{\text{ВЧ}}^* + w_2 \overline{K_{\text{ЛОК}}^*} + w_3 \overline{K_{\text{ЛОКНЧ}}^*}]. \quad (7)$$

Частные показатели в (7) рассчитываются следующим образом. СКО высокочастотной составляющей $\sigma_{\text{ВЧ}}$ оценивается по разности исходного изображения и оценки фоновой составляющей, полученной путем свертки с весовой функцией окна большой апертуры, после чего корректируется с учётом оценки СКО собственного шума $\sigma_{\text{ш}}$:

$$\sigma_{\text{ВЧ}}^* = \begin{cases} (\sigma_{\text{ВЧ}}^2 - \sigma_{\text{ш}}^2)^{0,5}, & \sigma_{\text{ВЧ}} > \sigma_{\text{ш}}, \\ 0, & \sigma_{\text{ВЧ}} < \sigma_{\text{ш}}. \end{cases} \quad (8)$$

Для оценки яркости фоновой составляющей в работе применена фильтрация исходного изображения пространственным фильтром с маской 128×128 пикселей. СКО шума $\sigma_{\text{ш}}$ оценивается по минимуму локального СКО в секторах $N \times N$ пикселей и умножается на поправочный коэффициент $w_{\text{ш}}$:

$$\sigma_{\text{ш}} = w_{\text{ш}} \min_{i,j} \{ \sigma_{\text{лок } ij} \}, \quad (9)$$

где $i = 1, 2, \dots, W/N, j = 1, 2, \dots, H/N$.

Из локальных СКО для предотвращения формирования заниженного значения СКО $\sigma_{\text{ш}}$ по (9) исключаются локальные оценки $\sigma_{\text{лок } ij}$, не удовлетворяющие хотя бы одному из условий:

$$p\sigma_{\text{лок } ij} < m_{\text{лок } ij} < 255 - p\sigma_{\text{лок } ij},$$

$$m_{\text{лок } ij} \neq 0, \quad m_{\text{лок } ij} \neq 255,$$

где $m_{ij \text{ лок}}$ – локальное математическое ожидание яркости.

Параметр p в работе принят равным 2,5. Необходимость введения дополнительных условий продиктована ограниченным диапазоном значений яркости сенсоров с глубиной цвета 8 бит: от 0 до 255. В работе оценка $\sigma_{\text{ш}}$ выполнялась в секторах 16×16 пикселей с $w_{\text{ш}} = 1,2$.

Весовой коэффициент $w(q)$, зависящий от глобального отношения сигнал-шум (ОСШ) $q = \sigma_{\text{вч}}^* / \sigma_{\text{ш}}$ учитывает степень искажения шумом малоразмерных деталей изображения. В работе он задан монотонно возрастающей сигмоидальной функцией $w(q) = 1 - \exp(-0,2q^2)$.

Среднее значение локального контраста $\overline{K_{\text{лок}}}$ определяется следующим образом:

$$\overline{K_{\text{лок}}} = \overline{LC_{ij}}, \quad (10)$$

где LC_{ij} – значение локального контраста в секторе с номером ij размером $N \times N$ пикселей, $i = 1, 2, \dots, W/N, j = 1, 2, \dots, H/N$, вычисляемое по (6).

Поскольку БГШ с нулевым математическим ожиданием увеличивает контраст (6) на $6\sigma_{\text{ш}}/255$, необходимо дополнительно выполнять коррекцию оценки локальных контрастов:

$$\overline{K_{\text{лок}}^*} = \overline{LC_{ij}^*}, \quad LC_{ij}^* = LC_{ij} - 6\sigma_{\text{ш}}/255. \quad (11)$$

Глаз, как оптический инструмент, имеет ограниченную разрешающую способность и в модели может быть представлен фильтром нижних пространственных частот. Этим объясняется то, что даже при низких отношениях сигнал-шум наблюдатель хорошо различает протяженные объекты на изображении [4, 8]. По этой причине в ИМПК также учитывается средний контраст низкочастотной (НЧ) составляющей изображения $\overline{K_{\text{локНЧ}}}$, которую в свою очередь получают кратномасштабным разложением (например, по базисным функциям Хаара). Поскольку НЧ составляющая кратности M имеет в $2M$ раз меньшие по сравнению с исходным изображением ширину и высоту, то локальный контраст LC_{ij} для подстановки в (11) оценивается только для соседних пикселей (по сектору 2×2 пикселя).

Коэффициенты w_1 , w_2 и w_3 определяют вес частных показателей в ИМПК (7). При проведении экспериментов автором приняты значения $w_1 = 0,5$, $w_2 = w_3 = 0,25$.

Результаты полунатурных экспериментов

Одной из областей применения безэталонных показателей качества является динамический выбор того или иного метода улучшения изображений и/или объединения кадров различных спектральных диапазонов [2, 3]. Поэтому при проведении экспериментальных исследований выполнялось сравнение ИМПК (7) с ИПК (3).

Ниже приведены результаты сравнения по двум тестовым изображениям: серого фона яркости 128 (рис. 1) и «Лена» (рис. 2) с наложенным на них аддитивным БГШ. Численные значения показателей качества сведены в таблицах 1 и 2 соответственно.

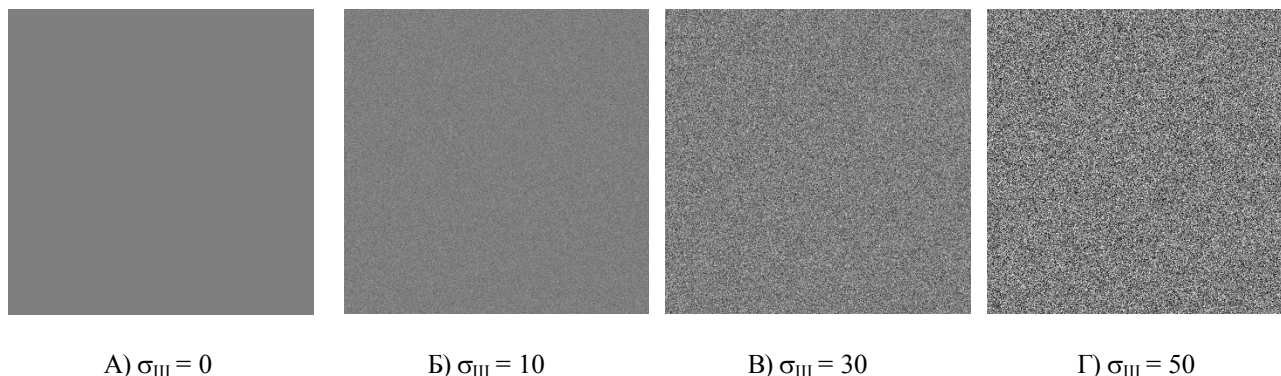


Рис. 1

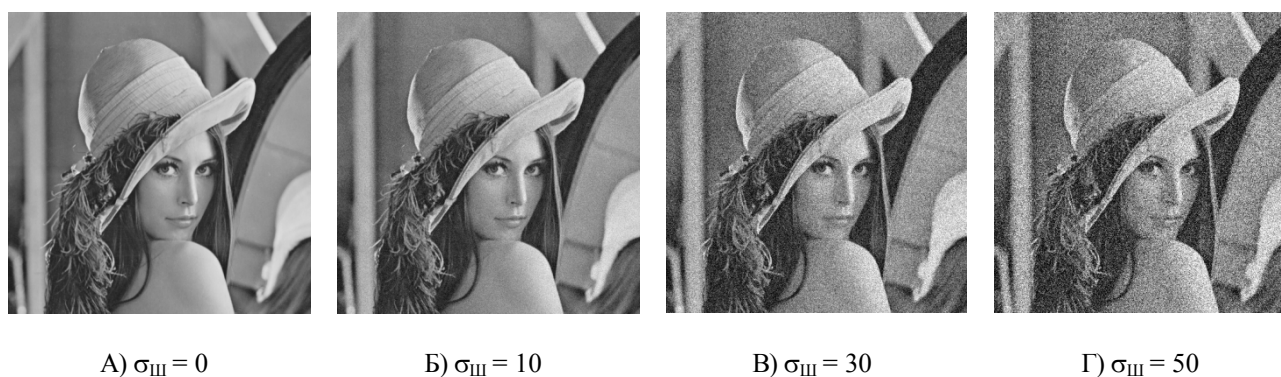


Рис. 2

Таблица 1

Q	$\sigma_{ш} = 0$	$\sigma_{ш} = 10$	$\sigma_{ш} = 30$	$\sigma_{ш} = 50$
$Q_{ИПК}$	0,3305	0,5394	0,8727	0,9928
$Q_{ИМПК}$	0	0,0088	0,0268	0,0276

Таблица 2

Q	$\sigma_{ш} = 0$	$\sigma_{ш} = 10$	$\sigma_{ш} = 30$	$\sigma_{ш} = 50$
$Q_{ИПК}$	0,9366	0,9887	0,9644	0,9125
$Q_{ИМПК}$	0,5697	0,5219	0,2511	0,0895

Из полученных численных значений видно, что при гауссовском распределении шума результаты оценки качества изображений с использованием безэталонного ИМПК приблизительно совпадают с субъективной оценкой их информативности.

Заключение

Таким образом, рассмотренный безэталонный нормированный ИМПК позволяет ранжировать цифровые изображения по критерию информативности (7). Показано, что с увеличением СКО шума значение ИМПК, в отличие от ИПК, снижается, что соответствует субъективному восприятию качества.

Библиографический список

1. Старовойтов В. В., Старовойтов Ф. В. Сравнительный анализ безэталонных мер оценки качества цифровых изображений // Системный анализ и прикладная информатика. 2017. № 1. С. 24-32.
2. Колчаев Д. А., Муратов Е. Р., Никифоров М. Б. Математическое обеспечение системы динамического выбора метода улучшения изображений в реальном времени // Известия ТулГУ. Технические науки. 2017. Вып. 2. С. 83-89.
3. Бондаренко А., Бондаренко М. Аппаратно-программная реализация мультиспектральной системы улучшенного видения // Современная электроника. 2017. № 1. С. 32-37.
4. Богданов А. П., Романов Ю. Н. Оценка качества цифровых изображений // Техническое зрение в системах управления – 2012: тезисы докладов. М.: ИКИ РАН, 2012. С. 218-226.
5. Воробель Р. А., Журавель И. М., Опыр Н. В., Попов Б. О., Дереча В. Я., Равлик Я. М. Метод количественной оценки качества рентгенографических изображений // Неразрушающий контроль и техническая диагностика: труды 3-й украинской научно-технической конференции. Днепропетровск, 2000. С. 233-236.
6. Конюхов А. Л., Костевич А. Г., Курячий М. И. Критерии оценки отношения сигнал/шум в активно-импульсных телевизионно-вычислительных системах // Доклады ТУСУРа. 2012. № 2. Ч. 1. С. 111-115.
7. Космическое земледелие: информационно-математические основы / под ред. Садовниченко В. А. М.: МГУ, 1998. 576 с.
8. Прэнт У. Цифровая обработка изображений: пер. с англ. Кн.1. М.: Мир, 1982. 312 с.

УДК 004.424; ГРНТИ 50.41.29

РАЗРАБОТКА ПРОГРАММНЫХ СРЕДСТВ ОБРАБОТКИ ЭЛЕКТРОКАРДИОСИГНАЛА С ПОМОЩЬЮ ТЕХНОЛОГИИ ИНФОРМАЦИОННОГО АНАЛИЗА

Н. Н. Гринченко, Е. С. Геращенко, В. Ю. Потапова

*Рязанский государственный радиотехнический университет,
Россия, Рязань, grinchenko_nn@mail.ru*

Аннотация. В данной работе описана реализация метода обработки электрокардиографического сигнала, предложенного профессором В.М. Успенским. Данный метод позволяет диагностировать некоторые заболевания, исследуя электрокардиограммы.

Ключевые слова. Портативный кардиограф, электрокардиографический сигнал, технология информационного анализа.

SOFTWARE DEVELOPMENT TO PROCESS ECS USING A TECHNOLOGY OF INFORMATIONAL ANALYSIS

N. N. Grinchenko, E. S. Gerashchenko, V. Y. Potapova

*Ryazan State Radio Engineering University
Russia, Ryazan, grinchenko_nn@mail.ru*

Abstract. This paper describes an implementation of method of ECS processing that was proposed by professor V. M. Uspensky. This method allows to diagnose some diseases using ECG.

Keywords. Portable cardiograph, ECS, technology of informational analysis.

Введение

Контроль деятельности сердечно-сосудистой системы является одним из важных аспектов своевременного обнаружения нарушений в организме. Для этой цели может быть применен портативный кардиограф, который позволяет осуществлять мобильную диагностику в домашних условиях: отслеживать изменения основных параметров сердечной дея-

тельности, делать предположения о возможных диагнозах и давать рекомендации о посещении специалиста. В данной работе рассматривается применение технологии информационного анализа по методу профессора В.М. Успенского, в совокупности с другими методами обработки электрокардиографического сигнала в составе программной части портативного кардиографа.

Модель портативного кардиографа

Разработанный портативный кардиограф имеет структуру, приведенную на рисунке 1.

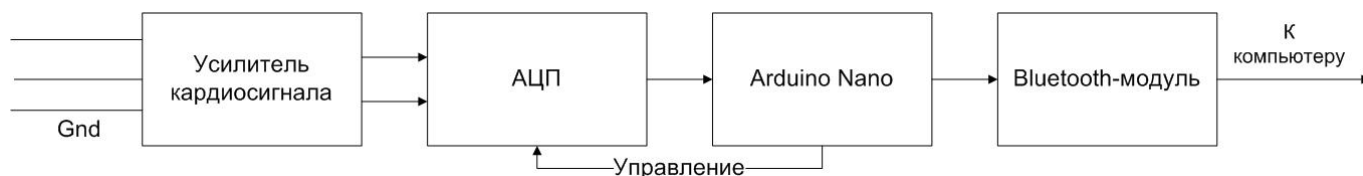


Рис. 1. Структурная схема устройства

Для съема ЭКГ предусмотрены три электрода, которые подключаются по первому стандартному отведению [1]. После съема сигнал поступает на аналоговую часть схемы, где выполняются его фильтрация и усиление. Далее сигнал передается на аналогово-цифровой преобразователь, управляемый микроконтроллером Arduino Nano, в результате чего формируется его представление с помощью дискретной функции. Передача на компьютер для вторичной обработки осуществляется через Bluetooth-модуль, образующий гальваническую развязку с компьютером и обеспечивающий защиту пациента от поражения электрическим током [2, 3].

Реализация технологии информационного анализа ЭКГ

Программная часть системы содержит набор последовательно работающих алгоритмов. Сначала выполняется модифицированный метод Пана-Томпкинса, позволяющий определить положение QRS-комплексов и их основных параметров [4-6]. На основе полученных данных работает алгоритм кластерного анализа формы комплексов, а затем алгоритм анализа аритмических событий [1]. Кроме этого, возможно построение графиков RR-интервалов, сохранение информации в базе данных для конкретного пользователя, а также формирование статистики изменения основных параметров сердечной деятельности за определенный промежуток времени.

Описанные методы выявляют только симптоматические данные, на основе которых можно сделать предположение о наличии определенных видов аритмий. Для того, чтобы сформировать другие вероятные диагнозы, необходимы дополнительные методы диагностики. В качестве такого метода в данной работе используется метод профессора В.М. Успенского.

Основными этапами информационного анализа по методу Успенского являются:

- 1) составление первичной кодограммы;
- 2) составление триграмм;
- 3) сортировка триграмм по убыванию частоты их появления и сопоставление результатов.

1. Составление первичной кодограммы

Для формирования последовательности необходимо вычислить три параметра каждого QRS-комплекса: знаки приращений амплитуд, RR-интервалов и углов, как показано на рисунке 2.

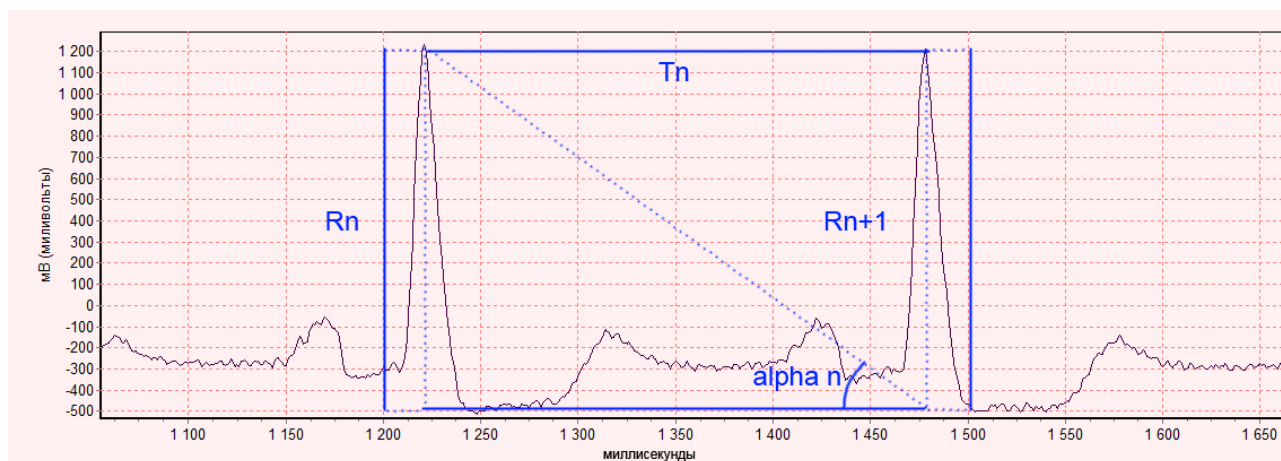


Рис. 2. Параметры QRS-комплекса

Для выделения таких параметров QRS-комплексов, как амплитуда и длительность RR-интервала, необходимых для вычисления перечисленных признаков, в данной работе используется модифицированный метод Пана-Томпкинса.

После вычисления трех параметров выполняется их оценка: если разность $n+1$ и n элементов положительна, то это значение кодируется как 1, если отрицательна или равна нулю – то как 0. Таким образом, QRS-комплекс описывается последовательностью из трех элементов, каждый из которых может принимать значение 0 или 1.

Далее по таблице 1 осуществляется замена набора признаков буквами.

Таблица 1. Правила кодирования

ΔR	ΔT	Δa	Код
1	1	1	A
0	0	1	B
1	0	1	C
0	1	0	D
1	1	0	E
0	0	0	F

2. Составление триграмм

На данном этапе метода ЭКГ описывается как первичная кодограмма (последовательность буквенных кодов). Далее необходимо составить из них триграммы, которые представляют собой набор трех буквенных кодов. Триграммы получаются путем смещения скользящего окна 1×3 по первичной кодограмме на один элемент от начала до конца. Пример представлен на рисунке 3.

Первичная кодограмма		
ACDDCDCADCCDFDAFCBAFDCECDFCC		
Наборы триграмм		
ACD CDD DDC DCD CDC DCA CAD ADC		
DCC CCD DCF CDF FDA DAF AFC FCB		
CBA BAF AFD FDC DCE CEC ECD CDF		
DFC FCC		

Рис. 3. Пример кодограмм и формирования на их основе триграмм

3. Сортировка триграмм по убыванию частоты их появления и сопоставление результатов

Когда сформирован набор триграмм, описывающих ЭКГ, нужно подсчитать количество появлений каждой триграммы и отсортировать их по убыванию частоты. Те наборы, которые окажутся в начале списка (наиболее часто встречающиеся) являются самыми значимыми для диагностики заболеваний. В методе выявлена закономерность, позволяющая однозначно сопоставить определенные триграммы конкретным диагнозам. В дальнейшем при анализе ЭКГ можно с большой степенью точности диагностировать болезнь, если соответствующая ей триграмма имеет достаточно большую частоту появления в описании. Среди заболеваний, которые возможно диагностировать данным методом, авторы приводят ишемическую болезнь сердца, вегетососудистую дистонию, сахарный диабет, различные виды гастрита, опухоли и другие [7].

Программная реализация метода

В данной работе описанный метод был реализован программно на основании источников [7, 8]. В качестве исходных данных использовались кардиограммы из базы данных The Long-Term ST Database, которая содержит ЭКГ длительностью от 21 до 24 часов с нормальными или искаженными ST-сегментами [9, 10]. База предназначена в основном для исследования ишемической болезни сердца. В аннотациях к ЭКГ указаны также другие заболевания, диагностированные у пациентов, в том числе сахарный диабет и гипертоническая болезнь.

В программе тестировались фрагменты ЭКГ длиной 10 минут, содержащие порядка 600 кардиоциклов. На рисунке 4а представлены сформированные триграммы в порядке убывания частоты появления для здорового человека, на рисунке 4б – для человека с сахарным диабетом, нарушением в коронарной артерии со стенокардией и с инфарктом миокарда в анамнезе, на рисунке 4в – для человека с нарушениями в коронарной артерии, стенокардией и гипертонией.

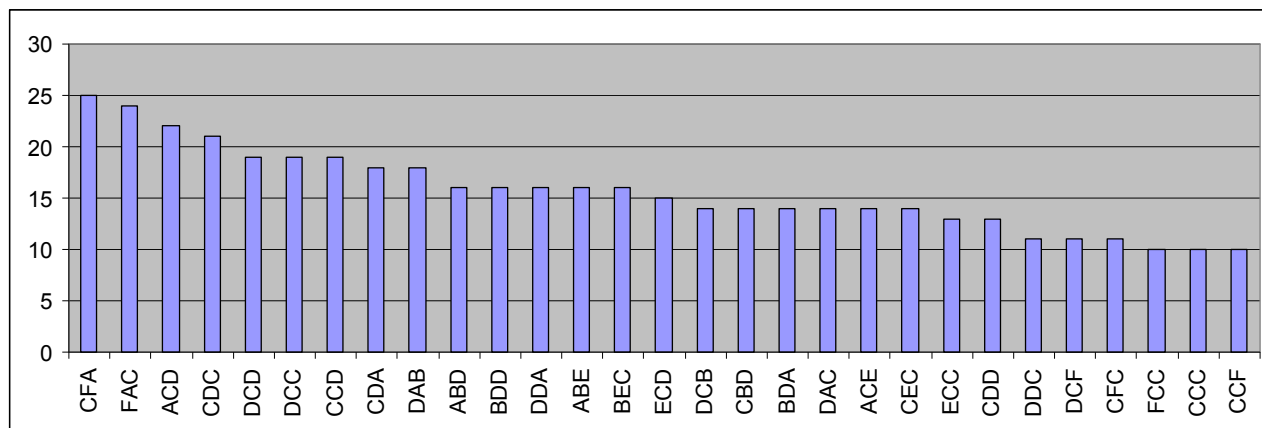


Рис. 4а – Гистограмма количества триграмм для ЭКГ здорового человека

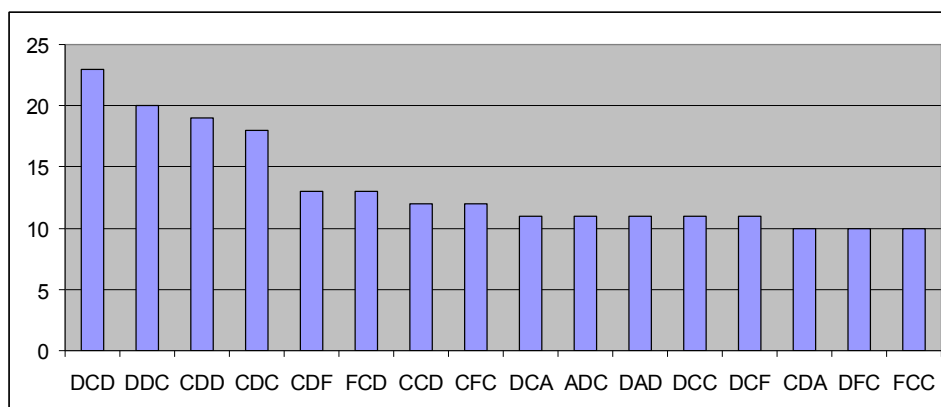


Рис. 4б – Гистограмма количества триграмм для ЭКГ человека с нарушениями

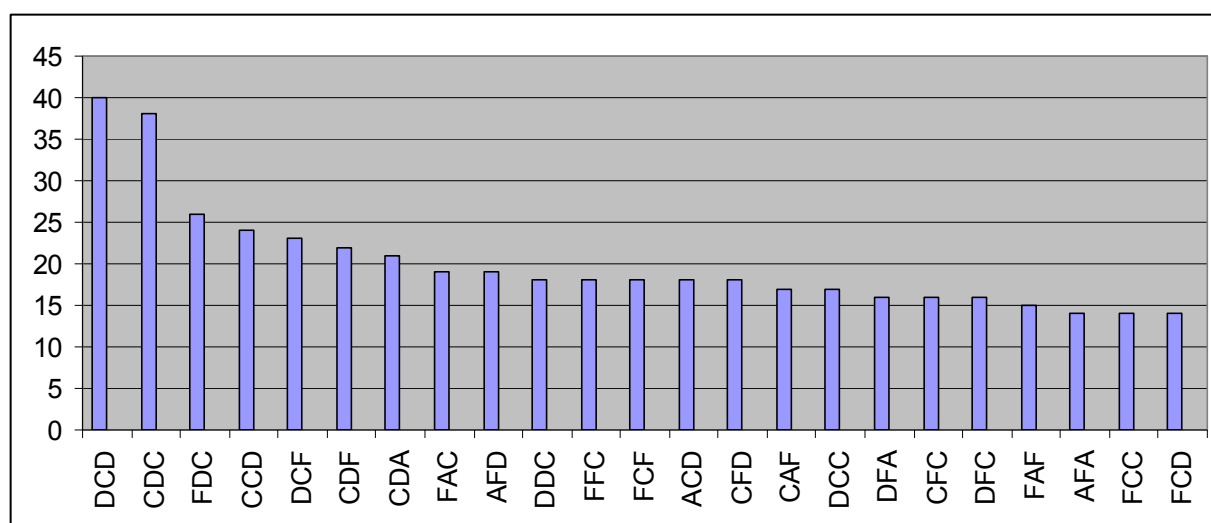


Рис. 4в – Гистограмма количества триграмм для ЭКГ человека с нарушениями

Полученные наборы триграмм для пациентов с сахарным диабетом и гипертонией не соответствуют эталонным триграммам, приведенным в работах [7, 8]. В данном случае, вероятно, это обусловлено тем, что приведенные заболевания – не единственные диагнозы пациентов, и другие нарушения оказывают большее влияние на результаты.

Выводы

Реализация метода информационного анализа ЭКГ с целью предварительной диагностики некоторых заболеваний в домашних условиях является целесообразным дополнением к существующему программному обеспечению портативного кардиографа. Возможна разработка собственных методов на основе описанного алгоритма и других подобных исследований.

Библиографический список

1. Барановский А.Л. Кардиомониторы. Аппаратура непрерывного контроля ЭКГ [Текст]: учеб. пособие для вузов / А.Л. Барановский [и др.]; под ред. А.Л. Барановского и А.П. Немирко. – М.: Радио и связь, 1993. – 248 с.
2. Гринченко Н.Н. Аппаратная реализация первичной обработки электрокардиосигнала [Текст]: Н.Н. Гринченко, Е.С. Геращенко, В.Ю. Потапова // Биотехнические, медицинские и экологические системы и комплексы. Биомедсистемы – 2016: материалы конференции. – Рязань, 2016. – С. 194-196.

3. Гринченко Н.Н. Регистрация и обработка электрокардиосигнала с применением устройства на основе операционного усилителя AD620 [Текст]: Н.Н. Гринченко, Е.С. Герашенко, В.Ю. Потапова // Методы и средства обработки и хранения информации: межвузовский сборник научных трудов. – Москва, 2017. – С. 4-6.
4. Pan J. A Real-Time QRS Detection Algorithm [Текст]: J. Pan, W.J. Tompkins // IEEE Transaction on biomedical engineering. – 1985. – Vol. BME-32. – № 3. – P. 230-236.
5. Леонова А.В. Модификация алгоритма распознавания QRS-комплексов в реальном времени Пана-Томпкинсона [Электронный ресурс]: А.В. Леонова, А.А. Агейченко // Инженерный вестник Дона. – 2015. – № 2. – URL: http://www.ivdon.ru/uploads/article/pdf/IVD_179_Leonova.pdf_25504bfd43.pdf
6. Герашенко Е.С. Применение модифицированного алгоритма Пана-Томпкинса для обработки нестационарного сигнала [Текст]: / Е.С. Герашенко // Новые информационные технологии в научных исследованиях; материалы XXII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. – Рязань. 2017. – С. 293-295.
7. Успенский В.М. Информационная функция сердца в диагностике заболеваний внутренних органов [Текст]: В.М. Успенский // Военно-медицинский журнал. – 2010. – № 9. – С. 45-52
8. Успенский В.М. Истоки информационной функции сердца и диагностической системы на основе информационного анализа электрокардиосигналов [Текст]: В.М. Успенский // В.М. Успенский // Технологии товаро-ведческой, таможенной и криминалистической экспертизы. Сборник научных трудов № 5 в 2 частях. – Санкт-Петербург, 2014. – С. 206-213.
9. Franc Jager, Alessandro Taddei, George B. Moody, Michele Emdin, Gorazd Antolic, Roman Dorn, Ales Smrdel, Carlo Marchesi, and Roger G. Mark. Long-term ST database: a reference for the development and evaluation of automated ischaemia detectors and for the study of the dynamics of myocardial ischaemia. Medical & Biological Engineering & Computing 41(2):172-183 (2003) [Электронный ресурс]: URL: <https://physionet.org/physiobank/database/ltstdb/lt03/>
10. Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23):e215-e220 [Circulation Electronic Pages; 2000 (June 13)] [Электронный ресурс]: URL: <http://circ.ahajournals.org/content/101/23/e215.full>

УДК 629.424; ГРНТИ 73.29.81

ПРОГРАММНЫЙ ИМИТАТОР ЭЛЕКТРОННОГО РЕГУЛЯТОРА ЧАСТОТЫ ВРАЩЕНИЯ ДИЗЕЛЬ – ГЕНЕРАТОРА 18-9ДГ

С. Гургуров, М.Б. Никифоров

*Рязанский государственный радиотехнический университет,
Российская Федерация, Рязань, nikiforov.m.b@evm.rsreu.ru*

Аннотация. В данной статье представлен один из возможных способов проверки и отладки программного обеспечения современного локомотива. В рамках способа рассмотрено взаимодействие управляющей программы тепловоза с электронным регулятором дизеля. Для этих целей разработан и применен программный имитатор электронного регулятора частоты вращения дизель – генератора, описание которого представлено в статье.

Ключевые слова. Тепловоз, микропроцессорная система управления, имитатор, электронный регулятор дизеля.

DIESEL LOCOMOTIVE CONTROL SYSTEM MODELLING

S. Gurgurov, M.B. Nikiforov

*Ryazan State Radio Engineering University,
Russian Federation, Ryazan, nikiforov.m.b@evm.rsreu.ru*

Abstract. This paper deals with questions of program software development and checkout for modern diesel locomotives using program imitator of diesel locomotive equipment. The program imitator embodiment for above mentioned purposes is carried out.

Keywords. Diesel locomotive, microprocessor control system, program imitator

Введение

Анализируя современные тенденции развития отечественного локомотивостроения за последние 20 лет, можно прийти к выводу, что при проектировании новых тепловозов, конструкторы придерживаются политики использования проверенных временем узлов и агрега-

тов. Данный факт увеличивает отказоустойчивость и повышает надежность локомотива в целом. Подтверждением тому является дизель – генератор 18-9ДГ с электронным регулятором частоты вращения дизеля (ЭРД) типа ЭРЧМ30ТЗ, используемый в целом ряде магистральных тепловозов серии 2ТЭ116, 2ТЭ116У, 2ТЭ25КМ и маневровых тепловозов серии ТЭМ7.

Несмотря на идентичность ЭРД, управляющие программы современных тепловозов отличаются между собой. Эти различия обусловлены конструктивными особенностями локомотивов и их модификаций.

Одной из основных функций ЭРД является выполнение директив микропроцессорной системы управления тепловозом (МСУ-Т). В связи с этим при разработке новых управляющих программ, или при изменении существующих разработчику необходимо иметь возможность анализа функционального взаимодействия МСУ-Т с ЭРД. Осуществить проверку работоспособности программного обеспечения можно непосредственно на локомотиве. Данный способ проверки эффективен, но весьма трудоёмок и затратен: вывод локомотива из эксплуатации, топливные расходы, необходимость в организации испытаний, включающая в себя, подготовку места для проведения испытаний, привлечение персонала, соответствующей квалификации.

Для исключения этих факторов предложен инновационный способ проверки работоспособности системы МСУ-Т в части взаимодействия с ЭРД. Данный способ заключается в использовании программного имитатора электронного регулятора частоты вращения дизель – генератора (ПИЭРД), алгоритмическое обеспечение которого позволяет максимально приближено описать функционирование ЭРД тепловоза.

Далее в докладе будет представлено описание ПИЭРД и способ проверки функционирования системы МСУ-Т с помощью ПИЭРД на стенде.

Описание программного имитатора электронного регулятора дизеля

ПИЭРД предназначен для моделирования процесса работы ЭРД локомотивов серии 2ТЭ116У и 2ТЭ25КМ. Представляет собой десктопное приложение для Windows XP и выше, внешний вид приложения изображен на рисунке 1, схема алгоритма работы ПИЭРД показана на рисунке 2.

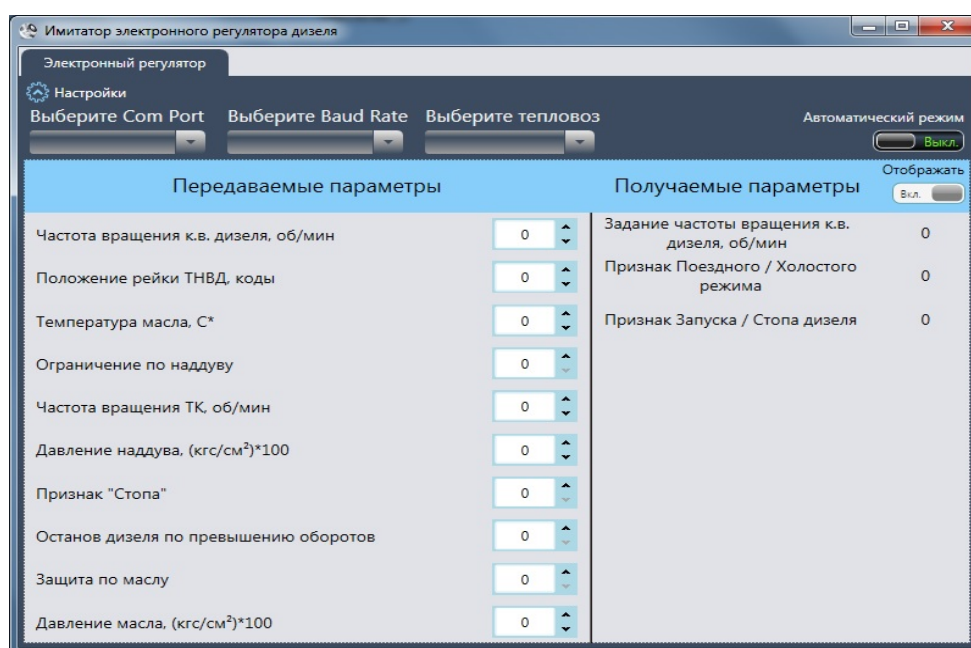


Рис.1. Внешний вид ПИЭРД

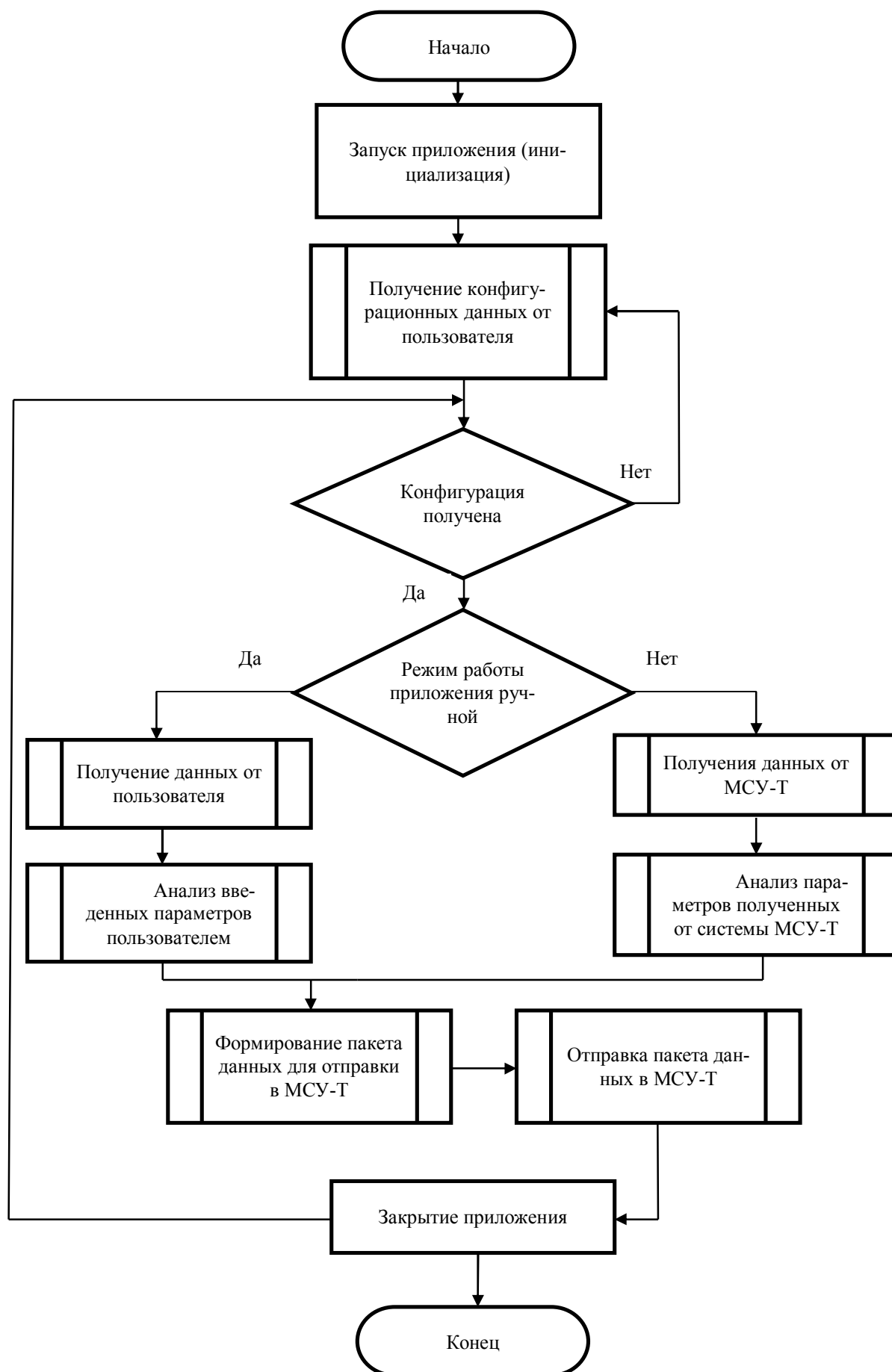


Рис. 2. Схема алгоритма работы ПИЭРД

Следует уточнить, что модель имитирует только необходимые процессы для корректной, совместной работы ЭРД с системой МСУ-Т соответствующих тепловозов. Имитатор позволяет произвести проверку правильности выполнения алгоритмов системы МСУ-Т в части задания для нагружения тягового генератора и обмена информацией с ЭРД. Так же, он позволяет воссоздать аварийные ситуации: признак «защита по давлению масла», «ограничение по давлению наддувочного воздуха», «останов по превышению предельно допустимых оборотов (разнос)», задание значений давления и температуры масла за пределы допустимых значений. Все эти аварийные ситуации, варьируемые в различных режимах работы тепловоза, позволяют довольно тщательно проверить работоспособность системы МСУ-Т в нештатных ситуациях.

ПИЭРД выполняет следующие основные функции:

- формирует и передает в систему МСУ-Т по соответствующему каналу связи информационный пакет, согласно установленному протоколу обмена данными между ЭРД и системой МСУ-Т (таблица 1);

- принимает и анализирует от системы МСУ-Т по соответствующему каналу связи информационный пакет, согласно установленному протоколу обмена данными между системой МСУ-Т и ЭРД (таблица 2).

ПИЭРД поддерживает два режима работы:

- ручной;
- автоматический.

В ручном режиме можно задать произвольное значение любого из параметров информационного пакета (таблица 2), т.е. сформировать произвольный информационный пакет и передать в систему МСУ-Т.

В автоматическом режиме формирование информационного пакета (таблица 2) происходит автономно. При формировании информационного пакета были учтены тонкости процесса нагружения тягового генератора системой МСУ-Т.

Во первых система МСУ-Т производит расчет заданной мощности исходя из соотношения:

$$P_{\text{полн.зад.}} = P_{\text{сел.}} + dP_{\text{рейки}},$$

где $P_{\text{сел.}}$ – нижнее фиксированное значение мощности;

$dP_{\text{рейки}}$ – рассогласование заданного значения выхода рейки с фактическим значением выхода рейки.

Во вторых система МСУ-Т производит расчет заданного напряжения на выходе выпрямительной установки.

$$U_{\text{зад.}} = P_{\text{полн.зад.}} / I_{\text{в.у.}},$$

где $I_{\text{в.у.}}$ – измеренный ток на выходе выпрямительной установки.

Для выполнения вышеперечисленных расчетов системе МСУ-Т необходимо получить следующие параметры: измеренное значение частоты вращения коленчатого вала (к.в.) дизеля, измеренный выход рейки топливных насосов высокого давления (ТНВД), напряжение и токи с шести независимых каналов на выходе выпрямительной установки (ВУ). Нужные значения выхода рейки ТНВД и частоту вращения к.в. дизеля задает ПИЭРД в автоматическом режиме работы. Фактически, им воссоздается идеализированная нагрузочная характеристика дизеля, без учета переходных процессов. Значения этих параметров представлены в таблице 3. Нужные значения напряжения и токов на выходе ВУ имитирует другой имитатор описание, которого выходит за рамки данной статьи.

Таблица 1. Информационный пакет система МСУ-ТП – ЭРД

№ байта	Наименование параметра
1	Задание частоты вращения коленчатого вала дизеля
2	Признак «Поездной/Холостой режим»
3	Признак «Работа/Стоп дизеля»

Таблица 2. Информационный пакет ЭРД – система МСУ-ТП

№ байта	Наименование параметра
1	Измеренная частота вращения коленчатого вала дизеля
2	Измеренный выход рейки ТНВД
3	Температура масла на входе в дизель
4	Признак «Ограничение по наддуву»
5	Измеренная частота вращения турбины турбокомпрессора
6	Давление наддува
7	Признак подтверждения команды «Работа/Стоп»
8	Признак «Останов по превышению оборотов (разнос)»
9	Признак «Защита по маслу дизеля»
10	Давление масла на входе в дизель

Таблица 3. Нагрузочная характеристика дизеля

Позиция контроллера машиниста	Частота вращения к.в. дизеля, п, мин-1	Мощность на выходе ВУ, Рсел, кВт	Заданный выход рейки, код.ед	Ориентировочный уровень полной мощности, Рполн, кВт	Ограничение приращения мощности, dP, кВт
1	350±6	120	620	120	0
2	470±6	220	685	220	0
3	590±6	325	765	325	0
4	680±6	415	1008	570	300
5	735±6	581	1140	780	300
6	770±6	700	1275	970	400
7	790±6	780	1320	1080	350
8	810±6	855	1389	1190	400
9	820±6	900	1400	1230	450
10	845±6	1010	1467	1360	450
11	875±6	1160	1560	1550	500
12	905±6	1330	1670	1740	500
13	940±6	1540	1770	1920	500
14	970±6	1700	1860	2080	500
15	1000±6	1900	2000	2230	500

Следует отметить, что имитатор при этом также задает нужные значения температуры и давления масла на входе в дизель, а так же измеренную частоту вращения турбины турбокомпрессора (ТК) и давление наддува. Эти параметры определены опытным путем, вследствие анализа осциллограмм реальных поездок.

Описание взаимодействия ПИЭРД с системой МСУ-Т

Имитация пуска дизеля. Пуск дизеля включает в себя 3 режима: «Прокачка топлива и масла» (далее «Прокачка»), «Прокрутка к.в. дизеля стартер – генератором» (далее «Прокрутка») и «Холостой ход». Для пуска дизеля на ЭРД должна поступить команда «Работа» от системы МСУ-Т. Следовательно, для имитации пуска дизеля на ПИЭРД так же должна поступить команда «Работа» от системы МСУ-Т. При ее отсутствии имитатор обнуляет ин-

формационный пакет (таблица 2) передаваемый в систему МСУ-Т, тем самым имитируя режим «Стоп» дизеля.

Имитация режима «Прокачка». При поступлении команды «Работа» ЭРД перед запуском дизеля выполняет прокачку топлива и масла. При этом система МСУ-Т контролирует наличие нужных значений давлений топлива на входе ТНВД (более $0,5 \text{ кг/см}^2$) и масла (более $0,3 \text{ кг/см}^2$) для запуска дизеля. При их отсутствии система МСУ-Т запрещает запуск дизеля. ПИЭРД выполняет имитацию предпусковых операций, следующим образом. Так как на имитаторе отсутствуют физически аппараты топливных и масляных насосов, то выполнение имитации предпусковых операций сводится к отсчету 60 секунд после получения команды «Работа» и установке требуемых значений давления топлива и масла в течение этого времени. Данное время отводится системой МСУ-Т для нагнетания насосами нужного давления топлива и масла необходимого для пуска дизеля. По истечению этого времени, при наличии соответствующих давлений топлива и масла, система МСУ-Т переходит к выполнению режима «Прокрутка».

Имитация режима «Прокрутка». После штатного завершения режима «Прокачка» процесс запуска дизеля переходит к режиму «Прокрутка». В этом режиме задается начальное вращение к.в. дизеля при помощи стартер – генератора. Для предотвращения разряджения аккумуляторной батареи на данную операцию выделяется определенное время (не более 12 секунд). Соответственно система МСУ-Т в течение этого времени ожидает выход дизеля на минимальные обороты, необходимые для запуска дизеля (более 250 мин^{-1}). Для этих целей в течении 5 секунд после завершения имитации режима «Прокачка» имитатор задает соответствующие значения выхода рейки ТНВД (более 150 ед.) и частоту вращения к.в. дизеля (более 250 мин^{-1}), имитируя прокрутку к.в. дизеля стартер – генератором.

Имитация режима «Холостой ход». После имитации режима «Прокрутка», система МСУ-Т считает дизель удачно запущенным, и посылает в имитатор признак «Холостой режим» и заданную частоту вращения к.в. дизеля равную 350 мин^{-1} . В ответ на это ПИЭРД посылает в систему МСУ-Т измеренную частоту вращения к.в. дизеля равную 350 мин^{-1} и измеренный выход рейки ТНВД равный 620 ед. Данные величины считаются нормальными для работающего дизеля на холостом ходу при нулевой позиции контроллера машиниста.

Имитация режима «Тяга». В режиме «Тяга» система МСУ-Т передает в ЭРД заданную частоту вращения к.в. дизеля и признак «Поездной режим». В соответствии с заданной частотой вращения ЭРД обеспечивает ее поддержание путем управления исполнительным устройством, выходной, вал которого соединен с рейками ТНВД. Имитатор же при получении признака «Поездной режим» сверяет, полученную от системы МСУ-Т, заданную частоту вращения к.в. дизеля с табличной (таблица 3). При совпадении значений ПИЭРД посылает в систему МСУ-Т информационный пакет, содержащий табличные значения совпавшей частоты вращения к.в. дизеля и соответствующие ей выход рейки ТНВД (таблица 3) и значения температуры и давления масла на входе в дизель, давление наддува и частоты вращения турбины ТК, что и является имитацией режима тяги.

При необходимости в любом вышеописанном режиме можно проверить целый ряд аварийных ситуаций, для этого достаточно перевести ПИЭРД в режим «ручной» и задать соответствующий параметр (таблица 2). По реакции системы МСУ-Т можно будет судить об ее корректной работоспособности.

ПИЭРД моделирует огромный спектр состояний дизеля, что позволяет имитировать основополагающие режимы работы локомотива. Этот факт дает возможность осуществлять проверку выполнения алгоритмов управляющей программы, как при ее доработках, так и на стадиях самой разработки без наличия локомотива.

Библиографический список

1. Филонов С.П., Гибалов А.И. Быковский В.Е. Тепловоз 2ТЭ116. – М.: Транспорт, 1985. – 328с.
2. Гургуров С.З., Никифоров М.Б., Сергеев С.В. Программный стенд для моделирования и отладки алгоритмического обеспечения системы управления тепловозом. Научный журнал «Известия Тульского государственного университета. Технические науки». 2017. Том №2. С. 41-47.
3. Гургуров С.З., Ким С.В. Имитатор электронного регулятора дизеля тепловоза 2ТЭ116У. Свидетельство №2017614013 от 05.04.2017.
4. Бабков Ю.В., Ким С.И. Система МСУ-ТП на тепловозе 2ТЭ116У. Журнал «Локомотив». 2009. №4. С. 14-19.
5. Сергеев С.В. Федоров М.В. Система МСУ-ТП на тепловозе 2ТЭ116У. Журнал «Локомотив». 2009. № 8. С. 15-18.
6. Тепловоз 2ТЭ116У. Руководство по эксплуатации. Часть 1: описание и работа. Луганск: ПАО «Луганск-тепловоз». 2007 – 149с.

УДК 004.93; ГРНТИ 28.23.15

АВТОМАТИЗАЦИЯ ИССЛЕДОВАНИЯ АЛГОРИТМОВ ОБРАБОТКИ ИЗОБРАЖЕНИЙ

С.А. Караев, Е.Р. Муратов, М.Б. Никифоров, М.А. Смыкова

*Рязанский государственный радиотехнический университет,
Российская Федерация, Рязань, nikiforov.m.b@evm.rsreu.ru*

Аннотация. В данной работе описана методика исследования алгоритмов обработки изображений, а так же программный стенд для автоматизации определения статистических характеристик алгоритмов.

Ключевые слова. Алгоритмы обработки изображений, программный стенд, предварительная обработка, пространственные фильтры

AUTOMATION OF IMAGE PROCESSING ALGORITHMS RESEARCH

S. A. Karaev, E. R. Muratov, M. B. Nikiforov, M. A. Smykova

*Ryazan State Radio Engineering University,
Russian Federation, Ryazan, nikiforov.m.b@evm.rsreu.ru*

Abstract. This paper describes the study of image processing algorithms, as well as software stand capable of automating it.

Keyword. Image processing algorithms, software stand, preprocessing, spatial filters

Введение

За последние десятилетия немаловажное значение получили алгоритмы обработки изображений, их используют во всех областях человеческой деятельности, таких как: медицина, наука, военная промышленность, картография и т.д. На исследование алгоритмов тратится большое количество времени, сил и средств. Но автоматизация смогла бы упростить и облегчить поиск наилучших способов обработки того или иного сюжета.

Классификация и характеристики алгоритмов обработки изображений

Обработка изображений включает в себя несколько стадий и первоначальной для всех изображений является предварительная обработка изображения, которая включает в себя:

- Алгоритмы направленные на улучшение цветовой передачи изображения называются *алгоритмами цветокоррекции*.
- Алгоритмы направленные на обработку пространственных характеристик изображений называются *алгоритмами пространственной фильтрации*.
- Алгоритмы выполняющие геометрические операции с изображением называются *алгоритмами геометрической обработки*.

- Алгоритмы выполняющие преобразования из одной цветовой системы в другую называются *алгоритмами цветопробразования*.

- Алгоритмы выделяющие на исходном изображении некоторые области по различным, часто неформальным условиям называются *алгоритмами сегментации*.

Далее при необходимости используют более узкоспециализированную обработку.

Все алгоритмы обработки изображений имеют свою характеристику и в качестве примера были использованы несколько самых популярных метода улучшения качества изображения, являющихся алгоритмами пространственной фильтрации.

1. *Линейное контрастирование* – производится линейное растяжение гистограммы обрабатываемого изображения. Пропорционально растягиваются все участки гистограммы, значения частот которых больше 5% от максимального значения на всей гистограмме. Расчет новой яркости для пикселя производится с использованием следующей формулы [1]:

$$I_{new}(x, y) = \frac{(I(x, y) - \Delta \min_j) \cdot L_j}{\Delta \max_j - \Delta \min_j}, j \in \overline{1, k},$$

где $I(x, y)$ – яркость пикселя входного изображения;

L_j – размер нового диапазона яркости для j -ого интервала (диапазона яркостей пикселей на гистограмме, подлежащий растяжению);

$\Delta \min_j, \Delta \max_j$ – левая и правая граница j -ого интервала на гистограмме исходного изображения;

k – количество интервалов.

2. *Эквализация гистограммы* - производится нелинейное растяжение гистограммы, при котором реальный яркостной диапазон исходного изображения (от минимального до максимального значения интенсивности) отображается на диапазон $[0, b]$. Математическая форма данного преобразования имеет вид[2]:

$$I_{new}(x, y) = \frac{\sum_{i=0}^{I(x, y)} H(i)}{\sum_i^b H(i)}.$$

3. *Медианная фильтрация* осуществляется следующим образом [3]:

$$B(x, y) = med\{M(x, y)\}.$$

Результат фильтрации есть медианное значение пикселей окрестности, форма которой определяется маской фильтра. Медианная фильтрация способна эффективно удалять из изображения помехи, независимо воздействующие на отдельные пиксели. Например, такими помехами являются "битые" пиксели при цифровой съемке, "снеговой" шум, когда часть пикселей заменяется на пиксели с максимальной интенсивностью, и т. п. Преимущество медианной фильтрации заключается в том, что "горячий" пиксель на темном фоне будет заменен темным, а не "размазан" по окрестности.

В большинстве своем характеристики зависят от сюжета и для определения и оценки характеристик, должно быть проведено большое количество опытов для разных сюжетов. Современные алгоритмы обработки изображений – это конвейеры, состоящие из последовательности более мелких (простых) алгоритмов. И итоговая характеристика конвейера обра-

ботки изображений зависит от алгоритма, включенного в конвейер. Как результат, всё это очень дорого, долго и сложно, наилучшим результатом будет автоматизация процесса. Главной задачей, на решение которой направлены исследования, является организация обработки данных отдельными алгоритмами в задаваемой последовательности, оптимизация данного процесса, а также получение и обработка статистической информации, по которой можно оценить работу исследуемых алгоритмов и провести их сравнительный анализ. В качестве программы, в которой реализована автоматизация исследования алгоритмов обработки изображений, представлен программный стенд.

Программный стенд

Программный стенд являет собой универсальное программное средство, позволяющее проводить отработку и сравнительный анализ алгоритмов обработки изображений, включая алгоритмы совмещения изображений различной физической природы и оценку качества получаемых совмещений в автоматическом режиме.

Стенд хранит разнородную информацию, которая используется в качестве входных данных, при этом данные тематически разделены по каталогам. В качестве данных могут использоваться видеопоследовательности, изображения, константы (числовые и строковые), навигационные данные сенсора, параметры сенсора (разрешение, углы обзора и т.п.), изображения виртуальной модели местности (ВММ). Алгоритмы, реализованные в виде отдельных функций, различные объекты и нестандартные форматы данных хранятся в пользовательских DLL-библиотеках. Архитектура программного стенда показана на рисунке 1.

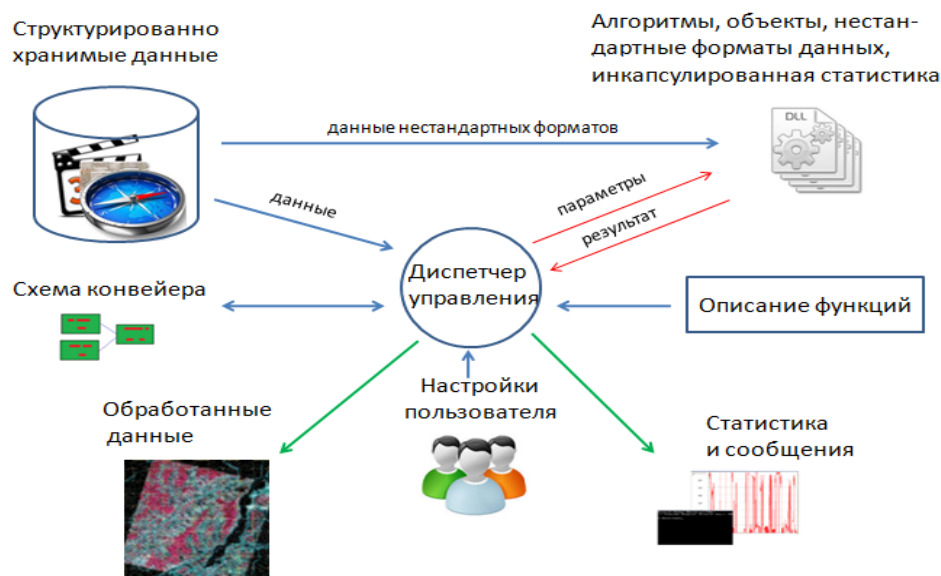


Рис. 1. Архитектура программного стенда

В качестве примера работы программного стенда можно привести результаты сравнительного анализа двух алгоритмов:

- Алгоритм выделения контуров на основе кластеризации состоящий из следующих процедур:
 - Предобработка входного изображения, включающая размытие с целью уменьшения уровня шума и растяжение гистограммы.
 - Fuzzy C-Means кластеризация для разбиения пикселей на два кластера. Начальные центры кластеров находятся с помощью алгоритма K-Means кластеризации.

3. Построение бинарного изображения A на основе разбиения, полученного на шаге 2.
4. Для устранения неоднородности больших сегментов, выраженной небольшими внутренними «дырками», применяется морфологическое закрытие бинарного изображения A структурным элементом B :
5. $A \cdot B = (A \oplus B) \ominus B$,
6. где $\oplus$ – операция дилатации, $\ominus$ – операция эрозии.
7. Оконтуривание полученного на предыдущем шаге изображения.
8. Векторизация полученных контуров.
9. Построение на результирующем изображении контуров, число точек которых превышает заданный порог с целью устранения малых незначимых объектов.
- Модернизированный метод совмещения ТВ изображения с ВММ изображением с использованием нечеткой кластеризации включает в себя:
 1. выделение контуров на телевизионном изображении с последующим совмещением контурного изображения с ВММ изображением с учетом или без учета линии горизонта;
 2. вызов функций выделения границ на основе нечеткой кластеризации из динамически подключаемой библиотеки, а также совмещение изображений для расширенного поля зрения.

Основной особенностью алгоритма совмещения с применением выделения границ на основе кластеризации является применение на этапе предварительной обработки (выделения границ) методов кластеризации.

Схемы конвейера стенда для оценки качества с применением алгоритмов на основе межпиксельного расстояния и на основе углов между линиями контуров приведены на рисунках 2 и 3 соответственно.

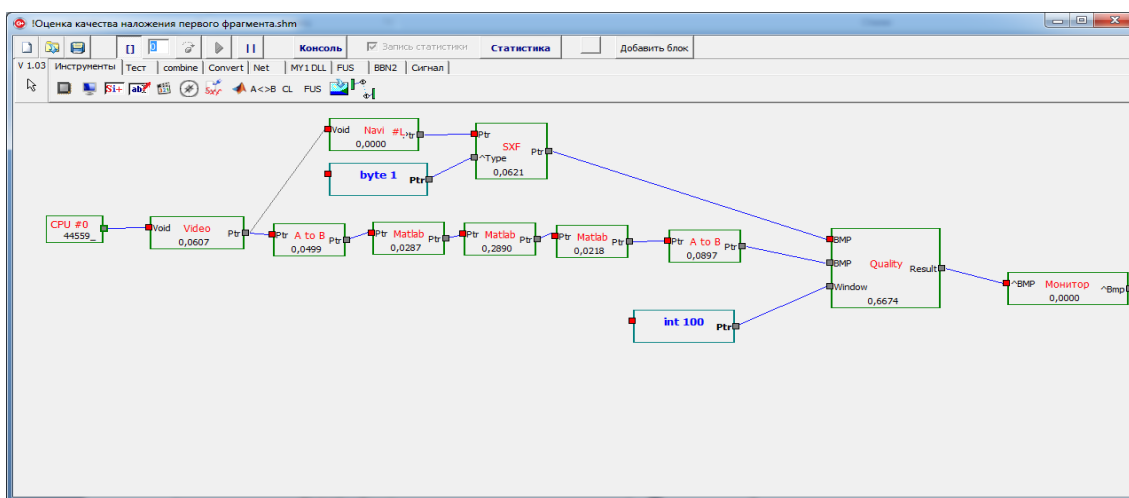


Рис. 2. Схема для оценки результатов с помощью алгоритма 1

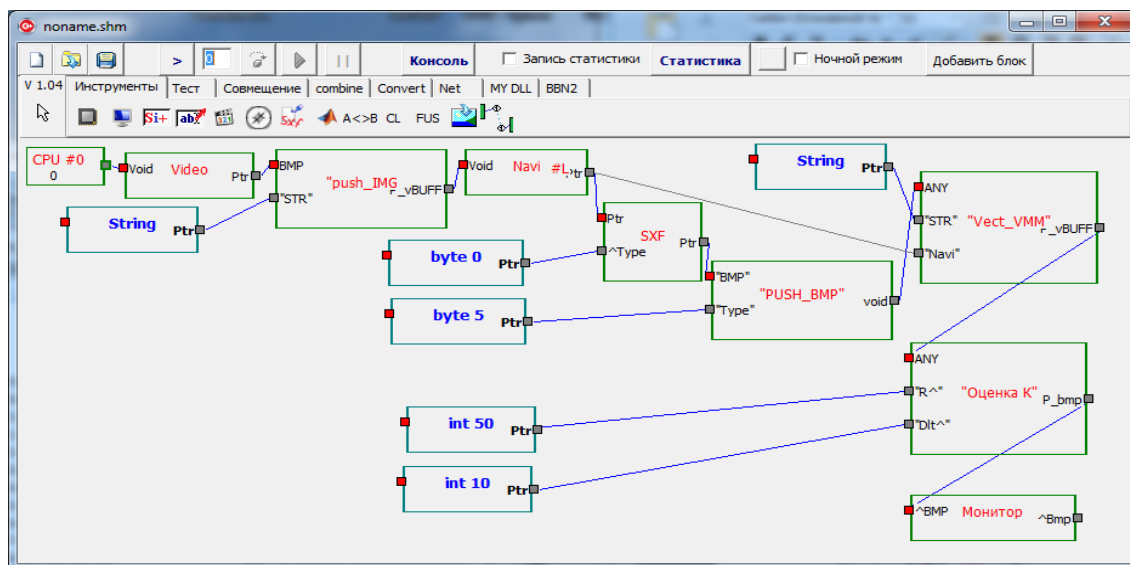
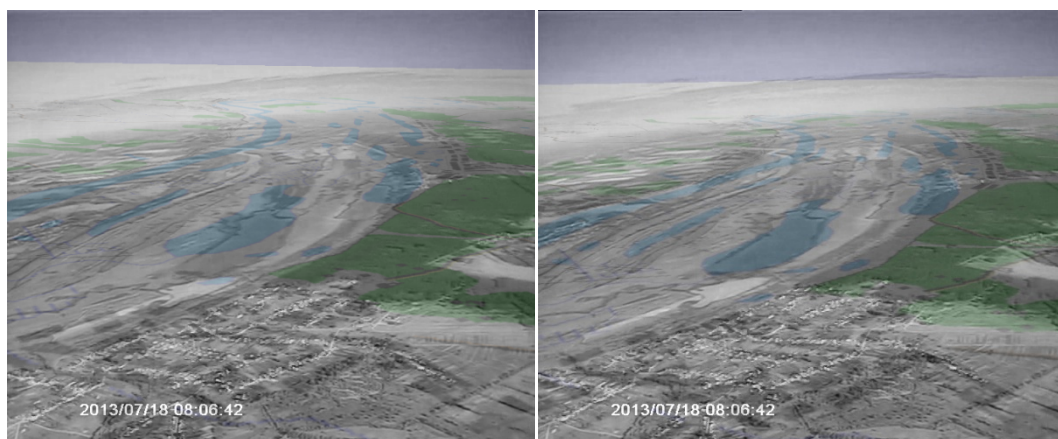


Рис. 3. Схема для оценки результатов с помощью алгоритма 2

На рисунке 4 а, б приведены изображения, соответственно до и после совмещения рассматриваемым подходом.



а) б)
Рис. 4. Результаты совмещения рассматриваемым алгоритмом

Алгоритм совмещения с применением выделения границ на основе кластеризации демонстрирует хорошие результаты совмещения для кадров, навигационные данные к которым содержат ошибки, не превосходящие предельно установленных.

Заключение

В статье рассматривается подход к автоматизации исследования алгоритмов обработки изображений, позволяющий сократить временные затраты проведение сравнительного анализа при выборе соответствующих алгоритмов. Приведен пример работы программного стенда для решения задачи выбора алгоритма совмещения изображений.

Библиографический список

1. Обработка изображений в авиационных системах технического зрения / под ред. Л.Н. Костяшкина, М.Б. Никифорова. – М.: ФИЗМАТЛИТ, 2016. – 240с. – ISBN 978-5-9221-1678-7.
2. А.И. Ефимов, Л.Н. Костяшкин, А.А. Логинов, Е.Р. Муратов, М.Б. Никифоров, А.И. Новиков. Обработка изображений в многоспектральных системах технического зрения. Вестник Рязанского государственного радиотехнического университета №2 (выпуск 60) 2017 г., стр.83-91.
3. K.A. Belyaeva, D.A. Kolchaev, A.A. Loginov. Statistical Methods of Analysis of Algorithms of Image Overlapping. of the 6th Mediterranean Conference on Embedded Computing (MECO). Montenegro, Bar, 2017, pp. 150-154.
4. Гонзалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. – 1072с.

УДК 004.8; ГРНТИ 28.23.37

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ BLOCKCHAIN В ЗАДАЧАХ АНАЛИЗА КАДРОВЫХ РЕСУРСОВ

А.Ю. Громов, А.А. Суслина, Д.А. Левшин

*Рязанский государственный радиотехнический университет,
Россия, Рязань, gromov.a.y@evm.rsreu.ru*

Аннотация. В работе рассматривается задача анализа информации в области управления человеческими ресурсами с оценкой достоверности обрабатываемых данных. Проводится исследование возможности применения технологии blockchain в качестве источника достоверной информации о кадровых ресурсах.

Ключевые слова. Block chain, HR-менеджмент, машинное обучение

APPLICATION OF BLOCKCHAIN TECHNOLOGY IN THE TASKS OF ANALYSIS OF HUMAN RESOURCES

A.Y. Gromov, A.A. Suslina, D.A. Levshin

*Ryazan State Radioengineering University,
Russia, Ryazan, gromov.a.y@evm.rsreu.ru*

Abstract. The article considers the task of analyzing information in the field of personnel management with an assessment of the reliability of the data being processed. The possibility of using block chain technology as a source of reliable information about human resources is investigated.

Keywords. Block chain, HR-management, machine learning

Введение

Рынок подбора кадровых ресурсов в последнее время переживает внедрение сразу нескольких новейших технологий.

- происходит расширение областей использования машинного обучения в задачах интеллектуального анализа данных, которые совсем недавно считались либо неразрешимыми с помощью подходов искусственного интеллекта, либо требовали несоизмеримых вычислительных затрат. Новые методы обучения, в том числе подходы с использованием глубокого обучения, за последние несколько лет показывают выдающиеся результаты в различных областях применения [1];

- в качестве инструментов проведения интервью начинают использовать робототехнику. Есть ряд пилотных проектов, в которых роботы-рекрутеры показывают результаты сравнимые с показателями реальных HR-менеджеров или близкие к ним;

- развивается технология blockchain, которой сулят повсеместное использование в качестве средства повышения достоверности информации различного рода.

Данная работа посвящена рассмотрению проблемы машинного обучения связанной с возможностью появления в исходных выборках данных недостоверной информации.

Алгоритм действий рекрутера

Алгоритм действий специалиста по подбору персонала регламентируется Приказом Министерства труда и социальной защиты РФ от 9 октября 2015 г. N 717н "Об утверждении профессионального стандарта "Специалист по подбору персонала (рекрутер)". На основе данного документа можно сформировать следующий набор действий [2]:

- поиск и привлечение кандидатов;
- отбор и оценка кандидатов;
- отбор и анализ информации о потенциальных кандидатах;
- оценка профессиональной квалификации кандидата;
- презентация вакансии кандидату;
- принятие решения о представлении кандидата работодателю;
- проверка рекомендаций с прежних мест работы на отобранных кандидатов.

Практически все выше описанные действия предусматривают обработку огромных объемов информации с заранее известными признаками, поэтому в последнее время все чаще прибегают к интеллектуальным методам анализа данных о соискателях и организациях.

С точки зрения достоверности информации есть несколько наиболее вероятных причин ее появления:

- информация зашумлена из-за дисперсии исходных признаков. Это является неотъемлемой частью истинных распределений случайных величин. При принятии экспертных решений используются специальные оценки, позволяющие контролировать влияние случайных процессов;

- требования к соискателям могут быть сформированы не на основе требований бизнеса. Сюда можно включить человеческий фактор, неправильно сформированную стратегию планирования человеческих ресурсов, стремление работодателей получить работников широкого профиля и другие причины;

- соискатели, при заполнении резюме, вносят недостоверные данные. Это может происходить как по причине стремления выделить свою кандидатуру из числа конкурсантов, так и по причине заблуждения о сути заполняемой информации.

Таким образом, перед рекрутером стоит ряд задач, в число которых входит оценка достоверности информации представляемой соискателем. К сожалению, проверка достоверности требует доступ к документальным источникам, которые зачастую недоступны рекрутеру, поэтому технология blockchain представляет большой интерес в контексте решаемой задачи.

Blockchain в анализе человеческих ресурсов

Blockchain, как технология распределенного хранения достоверных записей является если не общим решением множества проблем связанных с проверкой достоверности информации о соискателях, то, по крайней мере, его основой.

Использование данной технологии в области HR-менеджмента позволит:

- хранить достоверную информацию в распределенной форме;
- применять методы интеллектуального анализ данных с целью выявления неявных закономерностей как в характеристиках бизнеса, так и в признаках соискателей;
- повысить эффективность работы рекрутеров, за счет автоматизированной проверки резюме на предмет недостоверной информации;
- проводить исследования предметной области бизнеса;
- расширить области применения машинного обучения за счет использования эмпирических распределений исходных данных близких к истинным.

В данной работе не рассматриваются технологии реализации концепции blockchain. Вместо этого будем придерживаться того предположения, что информация, полученная из

источников, построенных по данной технологии считается достоверной с некоторой погрешностью. Погрешность появляется из-за динамики бизнеса, технологического развития общества и изменения знаний, умений и способностей соискателей.

Процесс подбора кадровых ресурсов

На основе алгоритма действия рекрутера и предположений о достоверности информации указанной в резюме можно сформировать процесс обработки заявок соискателей изображенный на рис. 1.

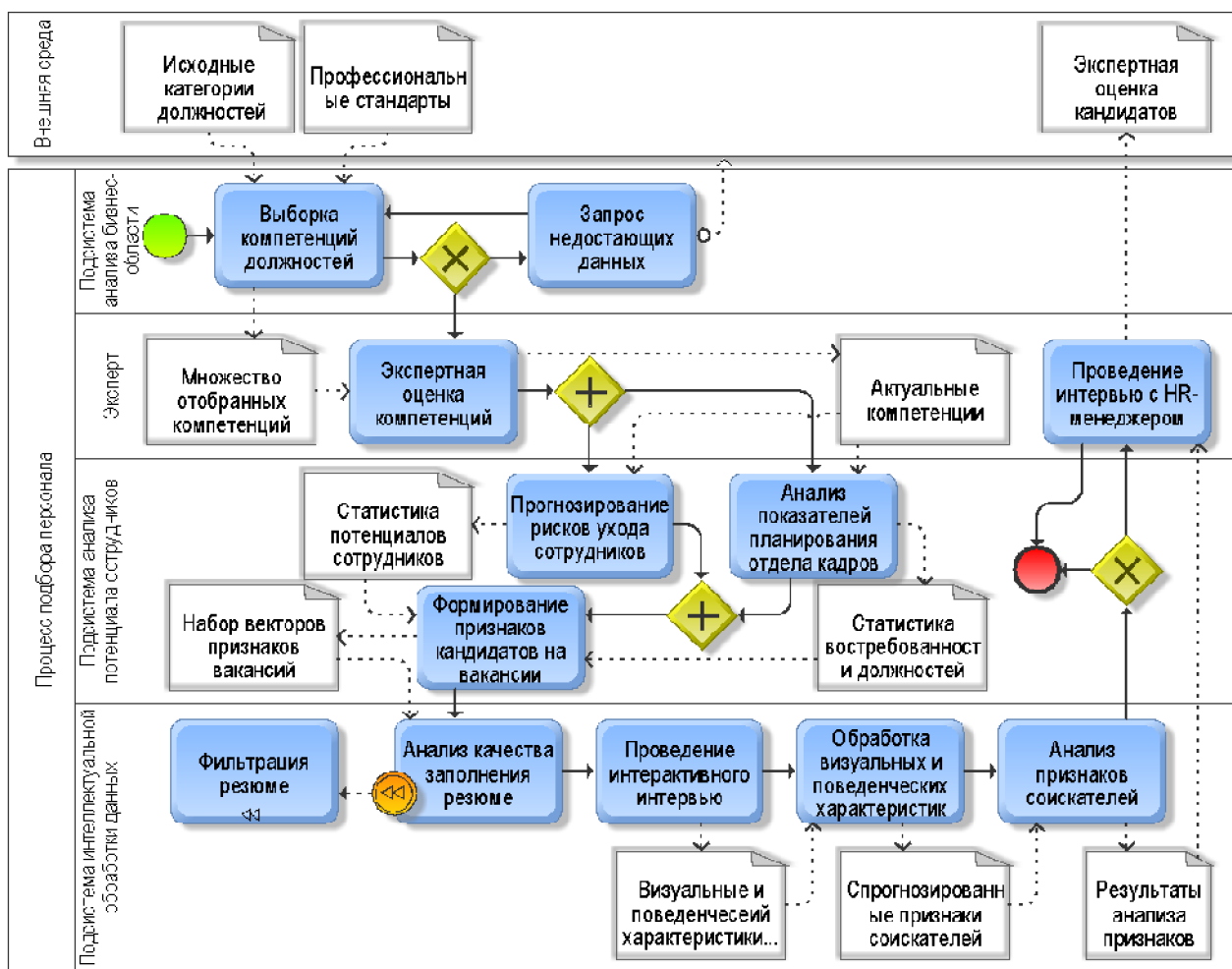


Рис. 1. Диаграмма процесса подбора кадровых ресурсов

В случае необходимости к процессу привлекается специалист по подбору персонала. Это может потребоваться при оптимизации входной информации и на этапе окончательного принятия решения о соответствии соискателя рассматриваемой вакансии.

Выборка компетенций должностей происходит на основе профессиональных и отраслевых стандартов, а так же требований бизнеса.

Экспертная оценка компетенций предназначена для их фильтрации и корректировки итогового множества. Для проведения данного этапа необходимо привлечение эксперта бизнес-области.

Прогнозирование риска ухода сотрудников, как и анализ планирования кадровых ресурсов предприятия, являются конкурсными задачами машинного обучения.

Анализ качества заполнения резюме так же использует методы машинного обучения, так как объем анализируемой информации повышает целесообразность использования подходов на основе нейронных сетей [1].

Обработка поведенческих и визуальных характеристик основана на анализе образов, почерка и видеоданных интервью соискателя.

Проведение интервью с HR-менеджером происходит при возникновении спорных ситуаций или при обработке приоритетных вакансий.

Выводы

Комплексная задача анализа человеческих ресурсов в настоящее время может быть решена с использованием подходов интеллектуального анализа данных. В случае повышения достоверности данных подаваемых на входы алгоритмов обучения их эффективность повышается, так как моделируемое распределение будет основываться на истинной информации. Методы выявления скрытых закономерностей данных, так же выигрывают от повышения достоверности. В качестве одного из средств повышения достоверности предлагается использовать технологию blockchain.

Предложенный в работе процесс обработки резюме соискателей позволит использовать достоинства методов прогнозирования и анализа, основанных на машинном обучении.

Библиографический список

1. Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучение / пер. с англ. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс, 2018. – 652 с.: цв. ил.
2. Профессиональный стандарт "Специалист по подбору персонала (рекрутер)" (Утвержден Приказом Министерства труда и социальной защиты РФ от 9 октября 2015 г. N 717н).

УДК 519.688; ГРНТИ 50.41.29

ИСПОЛЬЗОВАНИЕ ЯЗЫКА R ДЛЯ СТАТИСТИЧЕСКОЙ ОБРАБОТКИ МЕТЕОРОЛОГИЧЕСКИХ ДАННЫХ

Е. С. Герашенко, М. Б. Никифоров, В. Ю. Потапова

*Рязанский государственный радиотехнический университет,
Россия, Рязань, grinchenko_nn@mail.ru*

Аннотация. В данной работе описано применение методов статистического анализа к метеорологическим данным. С помощью многокритериальной регрессии и нелинейной регрессии получены уравнения для упрощения методов прогнозирования и определения текущих погодных условий, в частности ощущаемой температуры воздуха.

Ключевые слова. Статистический анализ, язык R.

USING R LANGUAGE FOR STATISTICAL PROCESSING OF METEOROLOGICAL DATA

E. S. Gerashchenko, M. B. Nikiforov, V. Y. Potapova

*Ryazan State Radio Engineering University
Russia, Ryazan, grinchenko_nn@mail.ru*

Abstract. This paper describes an application of methods of statistical analysis to meteorological data. The equations that simplify methods of forecasting and determination of current weather conditions were derived using multicriterion regression and non-linear regression.

Keywords. Statistical analysis, R language.

Введение

Необходимость статистической обработки экспериментальных данных часто возникает в самых различных предметных областях, таких, как метеорология, медицина, помехоустойчивое кодирование. Существует большое число инструментов обработки больших объемов данных с помощью методов статистической обработки. Один из них – язык R, который содержит большое количество встроенных методов и дополнительных пакетов.

В работе рассматривается применение языка R для анализа метеорологических данных, полученных с портативной метеостанции в течение года.

Модель портативной метеостанции

Разработанная портативная метеостанция имеет структуру, приведенную на рисунке 1.

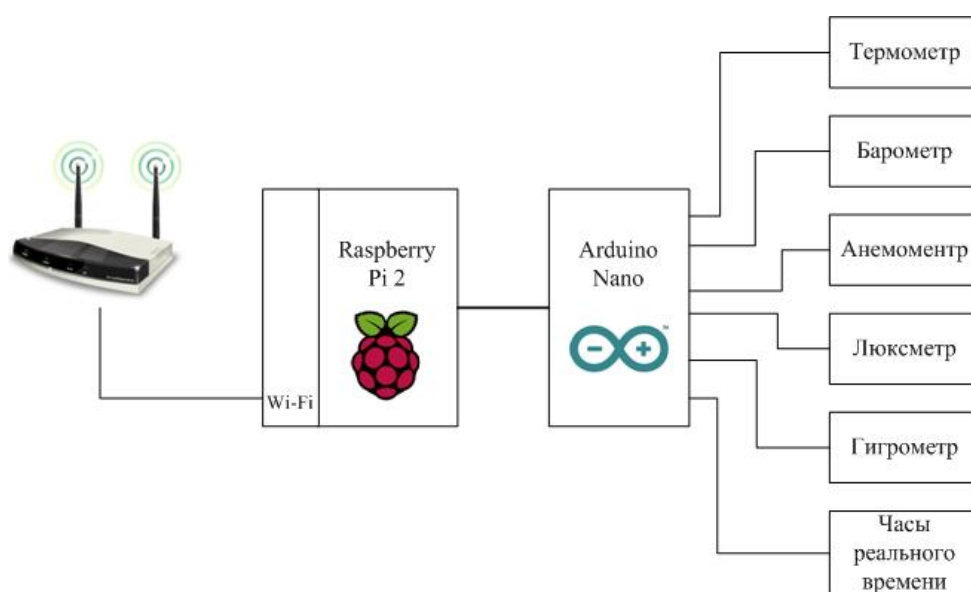


Рис. 1. Структурная схема устройства

В качестве устройств, получающих метеорологическую информацию, выступают бюджетные модели сенсоров. Данные от датчиков передаются на Arduino Nano, на котором осуществляется их проверка на корректность. В том случае, если значения находятся в допустимом диапазоне, данные отправляются на Raspberry Pi 2, на котором осуществляется их обработка. Если данные выходят за пределы допустимых диапазонов, устройство еще раз запрашивает погоду.

После обработки метеорологические данные передаются по Wi-Fi на роутер, с которого коммутируются и доставляются пользователю через веб-интерфейс.

В качестве методов обработки в данном устройстве выступают:

- 1) метод определения вероятности ночных заморозков;
- 2) метод определения ощущаемой температуры;
- 3) метод определения температуры и влажности воздуха на ближайшие сутки;
- 4) программная реализация погодного калькулятора Zambretti;
- 5) прогнозирование на основе понятия барической тенденции.

Статистической обработке подвергаются результаты метода определения вероятности ночных заморозков и метода определения ощущаемой температуры.

Многокритериальная регрессия для определения вероятности ночных заморозков

Метод определения вероятности ночных заморозков, предложенный профессором П. И. Броуновым, позволяет рассчитать вероятность на основе скорости изменения температуры в промежутке между 13 и 21 часами [1, 2]. Для расчета используются 6 линейных уравнений.

Метод многокритериальной регрессии использовался, чтобы спрогнозировать вероятность ночных заморозков, исходя из трех независимых переменных: времени года, температуры в 13 часов и температуры в 21 час [3, 4]. В качестве зависимой переменной выступала вероятность ночных заморозков. В табл. 1 представлен фрагмент исходных данных, при этом значения температуры получены с портативной метеостанции, а вероятность рассчитана по методу Броунова.

Таблица 1. Исходные данные для многокритериальной регрессии

Время года	Т в 13 часов	t в 21 час	Вероятность ночных заморозков
Зима	4	2	1
Зима	7	5	0.7
Весна	11	5	0.8
Весна	11	8	0.4
Лето	14	10	0.2
Лето	18	13	0
Осень	5	4	0.6
Осень	8	4	0.8

Код на языке R имеет следующий вид:

```
data=read.table("data.txt", header=TRUE, sep=",") # загрузка данных
model=lm(data$Prob~(data$Season+data$t1+data$t2)) # построение многокритериальной линейной модели
summary(model) # вывод основной информации
```

В результате построения многокритериальной линейной модели регрессии вероятность ночных заморозков можно рассчитать по уравнению следующего вида:

$$y = spr \cdot 0.005 - summ \cdot 0.058 + win \cdot 0.135 + t_1 \cdot 0.057 - t_2 \cdot 0.144 + 0.908,$$

где y – вероятность ночных заморозков;
 $spr, summ, win \in \{0,1\}$ – текущий сезон;
 t_1, t_2 – значения температуры.

Вероятность, рассчитываемая по приведенному уравнению, соответствует вероятности, полученной по методу Броунова, что делает данный метод применимым в составе портативной метеостанции. Таким образом, с помощью построения многокритериальной модели регрессии от 6 уравнений удалось перейти к одному более сложному.

Нелинейная регрессия для определения ощущаемой температуры

Метод расчета ощущаемой температуры на основе ветро-холодового индекса учитывает реальную температуру и скорость ветра и описывается следующей формулой:

$$T_o = T_p - av^b + cv,$$

где T_o – ощущаемая температура;

T_p – реальная температура;

v – скорость ветра;

a, b, c – некоторые коэффициенты, которые необходимо рассчитать.

Общий вид формулы был получен в результате анализа таблицы соответствия реальной температуры, ощущаемой температуры и скорости ветра из источника [5]. Так как данная зависимость является нелинейной, для определения коэффициентов необходимо построить нелинейную регрессионную модель. В качестве исходных данных используются значения скорости ветра, полученные с метеостанции, при реальной температуре 10°C, и значения ощущаемой температуры, определенные по таблице из источника [5]. В табл. 2 представлен фрагмент исходных данных.

Таблица 2. Исходные данные для нелинейной регрессии

Скорость ветра v	Ощущаемая температура T_o
0	10.000
1	10.000
2	9.914
3	9.784
4	9.620
5	9.430
6	9.218
7	8.985

Ниже приведен код на языке R.

```
test=read.table(«weather.txt», header=TRUE)
attach(test)
test.mod=nls(T~10-a*v^b+c*v,start=list(a=0.1,b=2,c=0.1))
```

В качестве параметров функции nls, которая строит нелинейную регрессионную модель, выступает уравнение, описывающее зависимость, и предполагаемые начальные значения исходных параметров [4, 6].

На рисунке 2 представлен вывод команды summary для построенной модели. Значения коэффициентов были получены за 18 итераций. На рисунке 3 представлен график, на котором точками отмечены экспериментальные значения в соответствии с таблицы 2, а кривая построена по полученному уравнению.

```

Nonlinear regression model
model: T ~ 10 - a * v^b + c * v
data: parent.frame()
a      b      c
0.20 1.28 0.20
residual sum-of-squares: 1.753e-19

Number of iterations to convergence: 18
Achieved convergence tolerance: 8.922e-06

```

Рис. 2. Параметры нелинейной модели

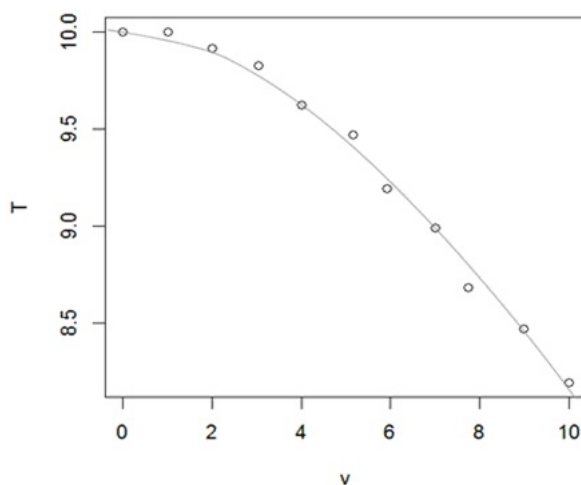


Рис. 3. Экспериментальные данные и полученная зависимость

Таким образом, ощущаемую температуру можно определить программно с использованием полученного нелинейного уравнения, а не анализа объемной таблицы.

Ощущаемая температура также может быть оценена с учетом влажности воздуха. В составе портативной метеостанции применяются оба метода, однако второй не рассматривался в рамках статистического анализа метеорологических данных.

Выводы

Применение методов статистического анализа к метеорологическим данным позволяет упростить существующие алгоритмы прогнозирования и определения текущих параметров погоды. Возможно интегрирование разработанных программ на языке R в состав ПО метеостанции для уточнения моделей на основе новых данных.

Библиографический список

1. Костров Б.В., Гринченко Н.Н., Потапова В.Ю., Тарасов А.С. Разработка сетевой версии системы сбора метео данных для прогнозирования ночных заморозков // Известие Тульского государственного университета. Технические науки. Выпуск 2017 С. 89-95.
2. Бабич Н.А. Лесная метеорология: Лабораторный практикум часть II // Н.П. Гаевский, Б.А. Мочалов. – Архангельск 1999. – С 35.
3. Мاستицкий С.Э., Шитиков В.К. Статистический анализ и визуализация данных с помощью R [Электронный ресурс]. – Режим доступа: <http://r-analytics.blogspot.com/>
4. Кабаков Р.И. R в действии. Анализ и визуализация данных в программе R // М.: ДМК Пресс, 2014. – 588 с.
5. Wind Chill [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Wind_chill (Дата обращения 19.12.2017)
6. Акинин М.В., Никифоров М.Б., Соколова А.В. Теория планирования эксперимента: методические указания к лабораторным работам // Рязань, РГРТУ, 2015. – 57 с.

УДК 004.932; ГРНТИ 28.23.15

ПРОГРАММНАЯ ИМИТАЦИЯ ПЯТЕН НЕРЕЗКОСТИ С ИСПОЛЬЗОВАНИЕМ КАРТЫ ГЛУБИНЫ

Д.А. Колчаев, С.В. Орлов

*Рязанский государственный радиотехнический университет,
Россия, Рязань, 3force3@mail.ru*

Аннотация. В данной работе рассматривается алгоритм программной имитации пятен нерезкости, использующий информацию о дальности до объектов формируемой по карте глубины, которая, в свою очередь, вычисляется по стереопаре. Приведен конвейер обработки с описанием основных этапов формирования результирующего изображения. Применение алгоритма позволяет получать изображения с имитацией пятен нерезкости на устройствах, оснащенных стереокамерой.

Ключевые слова. Пятно нерезкости, стереозрение, фильтрация изображений, интегральный показатель качества, боке.

SOFTWARE IMITATION OF BLURRED SPOTS WITH THE USE OF DEPTH MAP

D.A. Kolchaev, S.V. Orlov

*Ryazan State Radio Engineering University,
Russia, Ryazan, 3force3@mail.ru*

Abstract. In this paper, an algorithm for software imitation of blurred spots is presented, which uses information about the range of objects formed by the depth map, which is calculated by a stereopair. The processing pipeline with description of the main stages of the resulting image formation is presented. Application of the algorithm allows to obtain images with imitation of blurred spots on devices equipped with a stereo camera.

Keywords. Circle of confusion, stereo vision, image filtering, integral quality indicator, bokeh.

Современные тенденции развития программно-аппаратного обеспечения камер смартфонов направлены на получение качественных фотографий при различных условиях видимости. Разрабатываются и применяются новые технологии, основанные на использовании стереокамер, пленоптических сенсоров [1]. Одним из не менее важных направлений развития мобильных сенсоров является создание программно-аппаратных методов, позволяющих имитировать или полностью копировать функции профессиональных фотоаппаратов, например, формирование пятен нерезкости. Пятно нерезкости — искаженное изображение точки, возникающее вследствие дифракции света на оправках компонентов оптической системы (дифракционный предел), а также вследствие остаточных aberrаций. Части изображения, содержащие пятна нерезкости на художественных фотографиях, часто называют термином «боке». Этот эффект широко используется для портретных фотографий, где несущественные объекты на заднем плане размываются, чтобы привлечь зрительское внимание к основному объекту в сцене. Создание этого эффекта на смартфонах затруднительно в связи с отсутствием полноценных объективов, позволяющих изменять диафрагму.

Решением этой проблемы является использование программной имитации пятен нерезкости с использованием карты глубины. Карта глубины представляет собой матрицу, размер которой соответствует размеру изображения с камеры, и каждый элемент которой хранит смещение в пикселях одной и той же точки на двух изображениях от стереокамер. Наличие информации о смещении, а также параметров оптической системы позволяют вычислить реальное расстояние от точки обзора (положения фотоприемной матрицы) до объектов, попадающих в поле зрения камер.

Схема конвейера, позволяющего произвести имитацию пятен нерезкости, представлена на рисунке 1, и содержит следующие основные этапы: предобработка, вычисление карты глубины, фильтрация карты глубины, размытие изображения.

На этапе предобработки производится ректификация и улучшение входных изображений. Ректификация заключается в проецировании изображений на одну плоскость, кото-

рая проходит через точки, лежащие на оптических центрах фотоприемных матриц камер, и вектор нормали которой сонаправлен с векторами, лежащими на оптических осях камер.

Ректификация позволяет получить изображения, для которых выполняется условие, что соответствующие пиксели на правом и левом изображениях лежат на одной строке. Выполнение этого условия позволяет в дальнейшем облегчить нахождение смещения соответствующих точек.

Существует множество алгоритмов улучшения изображений, результат работы которых напрямую зависит от входных данных. Для обеспечения универсальности обрабатываемых данных предлагается использовать алгоритм автоматического выбора методов улучшения изображений [2]. Основная идея алгоритма заключается в выборе из некоторого множества различных методов улучшения наилучшего метода на основании интегрального показателя качества (ИПК).

Полученные обработанные изображения (стереопара) поступают на этап вычисления карты глубины. Вычисление значений карты глубины происходит за счет нахождения смещения пикселя левого изображения относительно этого же пикселя на правом изображении с использованием корреляционной оконной функции [3].

Применение корреляционной функции для нахождения соответствия между пикселями на левом и правом изображениях порождает ложные соответствия на участках, имеющих однородную интенсивность (стены, маркерная доска и т.д.), что приводит к неправильному определению значения дальности до объектов в карте глубины (черные области на изображении результата этапа вычисления карты глубины на рисунке 1).

Для решения проблемы неправильного вычисления дальности до объектов в отдельных областях применяются методы фильтрации изображений. Применение простых линейных фильтров, основанных на вычислении среднего в окне, применении функции Гаусса или Лапласиана, приводит к потере четкости границ на изображении карты глубины [4].

Для повышения качества фильтрации и сохранения границ применяется метод, основанный на использовании направляющего изображения, а также метод взвешенных наименьших квадратов [5,6]. Основная идея алгоритма заключается в использовании направляющего изображения (цветное или в градациях серого изображение с камеры), по которому производится расчет коэффициентов, минимизирующих функцию взвешенных наименьших квадратов (WLS), которые в дальнейшем определяют, каким образом будет происходить размытие карты глубины.

На последнем этапе происходит размытие изображения с камеры, учитывая значения в карте глубины, полученной на предыдущем этапе.

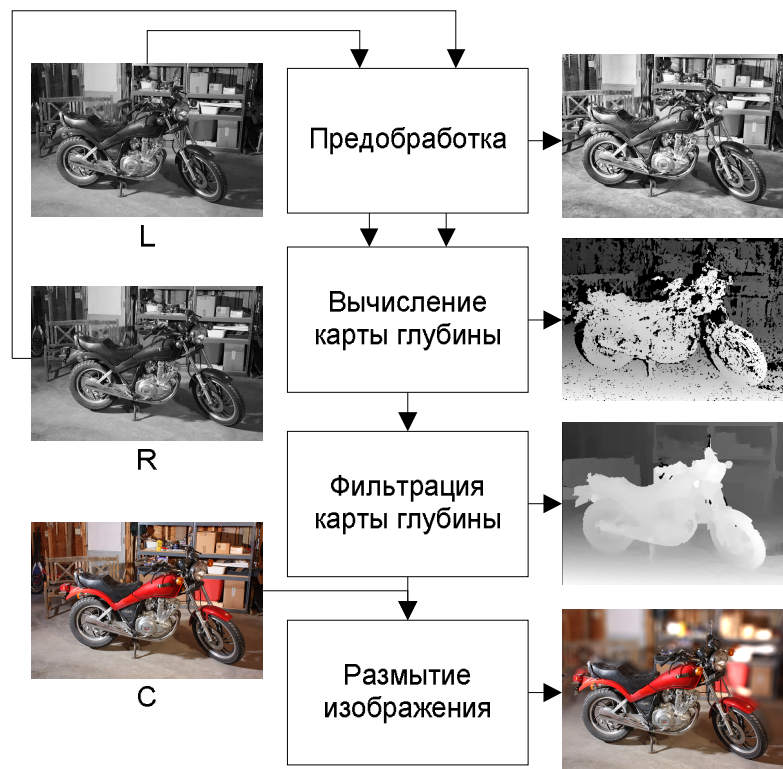


Рис. 1. Схема конвейера формирования пятен нерезкости на изображении

Обозначим I – как входное изображение с камеры, D – изображение карты глубины, тогда значения изображения F , содержащего результат имитации пятен нерезкости, будут формироваться следующим образом:

$$F(x, y) = I_b(x, y) \cdot (1 - v(x, y)) + I(x, y) \cdot v(x, y),$$

где I_b – размытое исходное изображение с имитацией пятен нерезкости;

$v(x, y)$ – весовой коэффициент, содержащий нормализованное значение карты глубины D ;

x, y – координаты пикселя на изображении.

Формирование изображения I_b производится по следующей формуле:

$$I_b(x, y) = \sum_{i, j \in S} (I(x-i, y-j) \cdot s(i, j)) / |S|,$$

где $s(i, j)$ – нормированная яркость пикселя используемого шаблона $s(i, j) \in [0, 1]$;

S – множество координат пикселей шаблона.

На рисунке 2 представлен результат работы программы имитации пятен нерезкости.



Рис. 2. Результат программной имитации пятен нерезкости

Шаблон позволяет задавать форму пятна нерезкости. В реальности на цифровых зеркальных фотоаппаратах, в зависимости от используемого объектива и диафрагмы, форма пятна нерезкости может изменяться. Для программной имитации используется заранее заданное шаблонное изображение пятна нерезкости. На рисунке 2 в нижней части изображены шаблоны, которые были использованы для формирования пятен нерезкости на изображениях.

Применение методов стереозрения позволяет вычислить карту глубины, которая задает степень размытия участков изображения.

Размытие осуществляется путем смещения исходного изображения в соответствии с заданным шаблоном, формирующим форму пятна нерезкости, и накопления нормализованной суммы яркостей. Получаемое в результате изображение схоже с изображением с цифрового зеркального фотоаппарата, при определенных значениях глубины резко изображаемого пространства, а также фокусного расстояния, выбранных для создания эффекта боке. Применение программной имитации пятен нерезкости не требует наличия объектива с возможностью

изменения диафрагмы, что позволяет применять этот метод в мобильных телефонах, оснащенных стереокамерой.

Библиографический список

1. M. Clapp and R. Etienne-Cummings, "A dual pixel-type imager for imaging and motion centroid localization." In Proc. ISCAS'01, Sydney, Australia, May 2001, vol. 2, pp. 501-504.
2. D.A. Kolchaev, Y.R. Muratov, M.B. Nikiforov, V.S. Gurov. Adaptive system of image processing. Proceedings of the 6th Mediterranean Conference on Embedded Computing. – Bar, Montenegro, 2017, pp. 141-144.
3. V.S. Gurov, D.I. Ustukov, Y.R. Muratov, M.B. Nikiforov. Implementing one of Stereovision Algorithms on FPGA. Proceedings of the 5th Mediterranean Conference on Embedded Computing (MECO). Montenegro, Bar, 2016, pp. 64-67.
4. R. C. Gonzalez and R. E. Woods. Digital Image Processing. Prentice Hall, 2 edition, 2002.
5. K. He, J. Sun, and X. Tang, "Guided image filtering," in Proc. Eur. Conf. Comput. Vis., 2010, pp. 1-14.
6. Dongbo Min, Sunghwan Choi, Jiangbo Lu, Bumsu Ham, Kwanghoon Sohn, and Minh N Do. "Fast global image smoothing based on weighted least squares". IEEE Transactions on Image Processing, 23(12)

УДК 004.932; ГРНТИ 89.57.35

ОПТИМИЗАЦИЯ АЛГОРИТМА ДВОЙНОЙ ИНТЕРПОЛЯЦИИ В ЗАДАЧАХ АНАЛИЗА ВЫСОТНЫХ ДАННЫХ

М.В. Никифоров, А.С. Тарасов

*Рязанский государственный радиотехнический университет,
Рязань, vb2005@yandex.ru*

Аннотация. В данной работе рассматриваются варианты ускорения работы алгоритма двойной интерполяции при обработке неоднородно распределённых данных. В рамках статьи рассмотрено применение различных интерполяционных функций двух переменных и проанализировано качество восстановления результата.

Ключевые слова. интерполяция, высотные данные, SRTM, карты

THE DOUBLE INTERPOLATION OPTIMIZATION IN THE ELEVATION DATA ANALYSIS

M.B. Nikiforov, A.S. Tarasov

*Ryazan State RadioEngineering University,
Ryazan, vb2005@yandex.ru*

Abstract. In this paper considers the development of a project module "Mobile GIS", providing the possibility of building surfaces and contours on the basis of a set of information about the altitude in certain points. The variants of the acceleration of the operation of the double interpolation algorithm for the processing of heterogeneously distributed data are considered. The application of various interpolation functions of two variables is considered and the quality of recovery of the result is analyzed.

Keywords. Interpolation tasks, elevation data, SRTM, maps

Введение

При составлении тематических карт часто приходится сталкиваться с проблемой сопоставления некоторому облаку точек A_n (набору данных, полученных в ходе проведения измерительных работ, представимых в виде трёхмерной точки $A_i = (x_i \ y_i \ z_i)$, где x_i и y_i – координаты в которых получено измерение, а z_i – численное значение измеренного параметра) некоторой функции $z = F_A(x, y)$. При этом, в большинстве случаев, можно говорить о том, что функция – кусочная и вряд ли может быть описана математически за адекватное время.

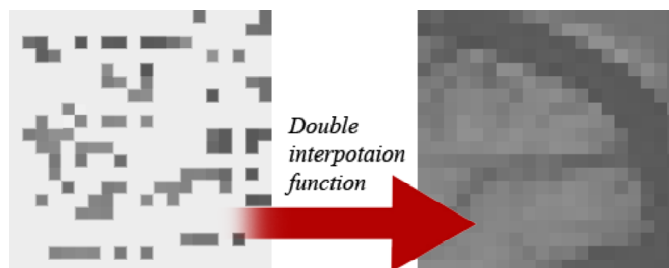


Рис. 1. Общий вид задачи двойной интерполяции

В этом случае оптимальным решением будет нахождение не самой функции, а требуемого для анализа набора значений. Способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений принято называть интерполяцией [1]. Двойная интерполяция – подвид интерполяционной задачи, где значение промежуточной величины вычисляется для функции двух переменных. В дальнейшем в статье речь пойдет именно о ней.

Двойная интерполяция может быть применена для получения множества значений любых параметров (данные о погоде, геологическое строение почвы и т.д.), в рамках данной статьи речь пойдет о нахождении высот на основе лидарной съёмки, барометрических измерений и данных SRTM [2].

Принципы построения матриц значений

Построение матрицы высот выполняется следующим способом: вычисляются или выбираются границы матрицы и шаг регулярной сетки. Для каждой точки сетки, где отсутствуют значения, выполняется двойная интерполяция.

Шаг сетки выбирается таким образом, чтобы обеспечить максимальную информативность, и при этом избежать вычисления избыточного количества значений. Границы области вычисления значений выбираются таким образом, чтобы в окрестностях точек было достаточно данных для вычисления адекватного значения.

К выбору границ области интерполяции следует подходить очень внимательно. В том случае, если она будет выбрана слишком маленькой, то далеко не для всех мест на карте удастся рассчитать значения, если будет слишком большой – то, несмотря на то, что значения были получены, они будут обладать очень низкой степенью адекватности. В ходе проведения эксперимента была выявлена следующая зависимости СКО от расстояния до ближайших точек, в которых возможно рассчитать значение. Следует отметить, что значения СКО весьма условны, и зависят от поведения функции на плоскости. Однако общий вид зависимости соответствует следующему графику (рис. 2):

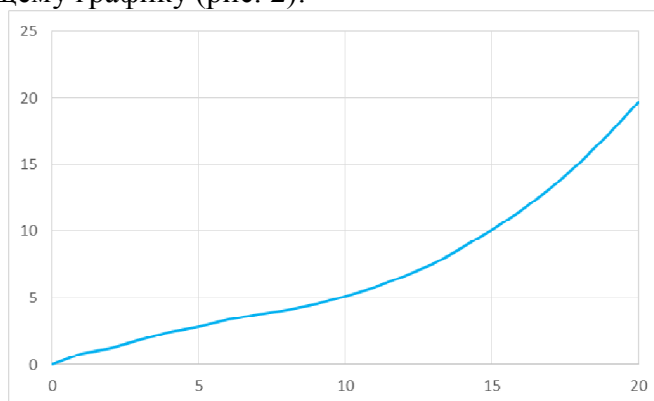


Рис. 2. Зависимость СКО от расстояния до пикселя

Задача интерполяции значений

Для каждой точки матрицы высот на основе двойной интерполяции выполняется расчёт значения следующим образом:

1. Для выбранной точки находятся n -ближайших значений.
2. На основе выбранных значений выполняется фильтрация и сортировка
3. К оставшимся значениям применяется интерполирующая функция.

Выборка ближайших значений к точке, может быть основана на переборе всех существующих точек. Однако применение данного варианта нецелесообразно, потому как при работе с большим количеством точек время обработки многократно возрастает. В качестве решения данной задачи были рассмотрены два варианта поиска соседних значений.

Поиск по матрице

В том случае, если все существующие значения уже представляют собой узлы в итоговой матрице, то допускается вариант спиралевидного поиска.

Алгоритм поиска следующий:

- Для выбранной точки изучается окрестность 3×3 на наличие в них исходных значений.
- В том случае, если собрано достаточное количество значений – работа алгоритма прекращается, и противном случае изучается следующая окрестность – 5×5 , и исключением из неё окрестности 3×3 (получается обход по периметру квадрата).
- Данный алгоритм выполняется до тех пор, пока не будет собрано достаточное количество точек n , либо же будет достигнута граница области поиска.

Данный алгоритм позволяет в среднем в 100 раз быстрее находить соседние значения в матрице, однако он применим только в том случае, если координаты исходного облака точек дискретны и уже представляют собой узлы конечной матрицы [3].

Поиск на основе разбиения на квадраты

Данный вариант является одним из способов индексирования значений в матрице. Присвоение индекса квадрату выполняется аналогично рисунку 3.

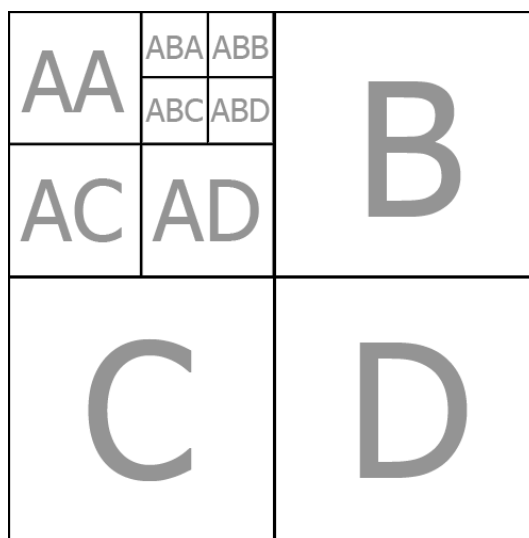


Рис 3. Разбиение матрицы на квадраты

Вся область изображения дробится на 4 квадрата, условно называемых А, В, С и D. Будем считать, что все точки, с координатами x меньшими, чем центр изображения и координатами y меньшими, чем центр изображения, принадлежат квадрату А. Аналогично для В, С, D. Для каждого квадрата определяется индекс в под квадрате.

Для каждого элемента из облака точек ставится в соответствие свой индекс, основанный на расположении точки. Например, для точки, расположенной в верхнем правом углу будет присвоен индекс «АААА». Степень дробления матрицы (количество букв) выбирается таким образом, чтобы каждый квадрат имел в среднем не более 10 точек. Такое разбиение позволяет наиболее эффективно находить соседние элементы.

Следует отметить, что применение подобной метода, в отличие от предыдущего, позволяет хранить координаты облака точек не только в целых, но и в дробных координатах. Кроме того, в матрицах с очень редким расположением элементов этот метод значительно повышает быстродействие в сравнении с предыдущим алгоритмом.

В том случае, если необходимо найти соседние элементы – анализируются значения в текущем квадрате. Если расстояния до этих элементов больше, чем до границы квадрата – то подключаются к анализу соседние квадраты, и так далее, пока не будет собрано достаточное количество значений.

Для хранения данных применяется класс Dictionary – словарь, в котором индекс – это строковый параметр, координата точки согласно описанному ранее принципу, а значение индекса – набор точек, соответствующих данному квадрату – рисунок 4 [4]. В квадрате может быть как несколько точек, так и ни одной.

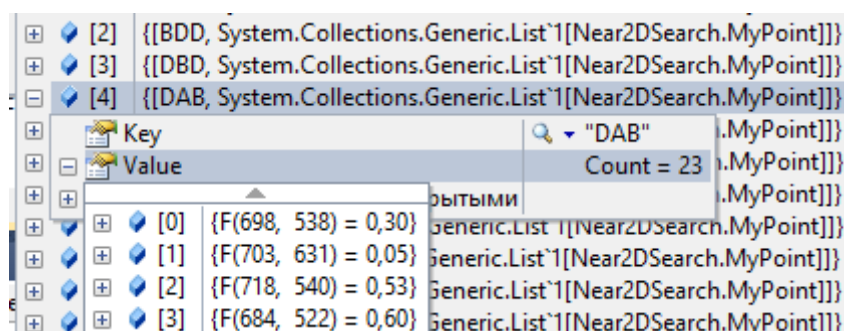


Рис. 4. Словарь с квадратами, и первые 3 точки для квадрата «DAB»

Поиск и применение интерполирующей функции

При решении задачи построения матрицы значений из облака точек важным фактором является выбор функции расстояния. Она устанавливает связь между интенсивностью цвета и расстоянием до пикселя.

$$W = \sum_{i=0}^n w_i * F_l(l_i) / \sum_{i=0}^n F_l(l_i).$$

где w_i – интенсивность цвета в i -ой точке;
 l_i – расстояние от определяемой точки до i -той;
 $F(l_i)$ – интерполирующая функция.

Вид функции расстояния подбирался вручную на основе изменения СКО в различных случаях. Наиболее достоверно себя ведёт функция $F_l(l) = l^{-4}$. Следует отметить, что при данных иного характера (например, изучение погодных явлений, анализ Земной коры и т.д.) вид функции может быть несколько иным.

Оценка качества восстановления изображения

Для того, чтобы оценить эффективность работы алгоритма была выбрана серия снимков SRTM. На изображениях было удалено определённое количество пикселей m (в %), после чего они были подвергнуты применению алгоритма двойной интерполяции (рис. 5). Полученное изображение сравнивалось с исходным на основе характеристики СКО.

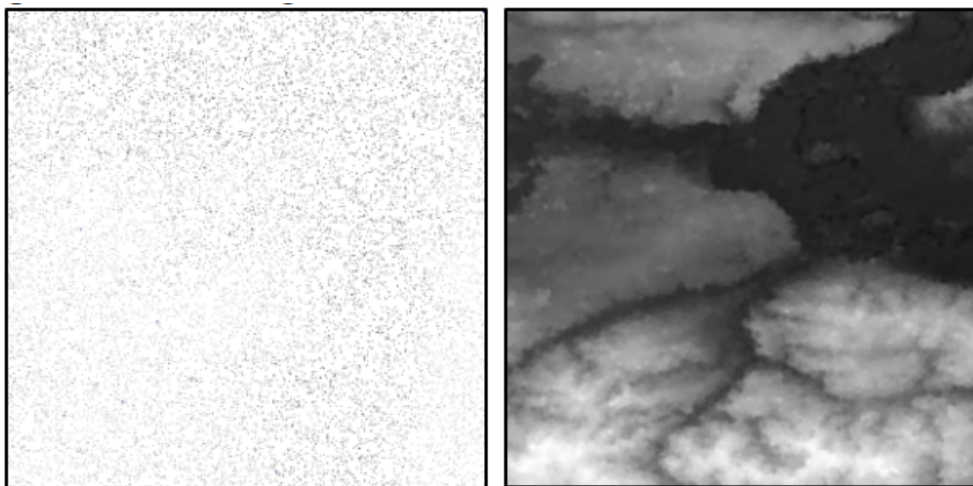


Рис. 5. Восстановленное изображение из 5% точек. СКО – 6,2.

Для того, чтобы проверить способность к интерполяции как при ничтожно малом количестве данных, так и при высоком, были выбраны следующие значения m : 0.01; 0.1; 0.5; 1; 2; 5; 20; 50; 75; 90; 99. Для каждого из них было выполнено 1000 экспериментов на разных наборах изображений. Результаты получились следующие (рис. 6):

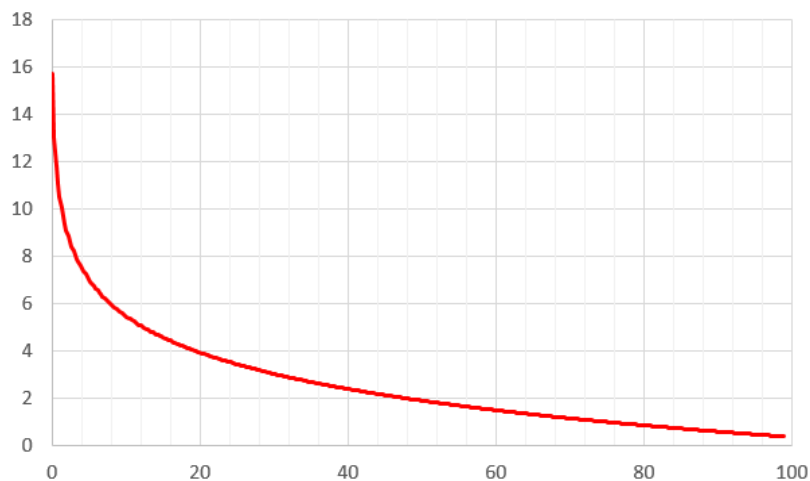


Рис. 6. Зависимость СКО от количества пикселей (в %)

Помимо задач интерполяции, данный алгоритм позволяет с высокой степенью эффективности сжимать данные и восстанавливать потерянные фрагменты изображений - рисунок 7 [5].

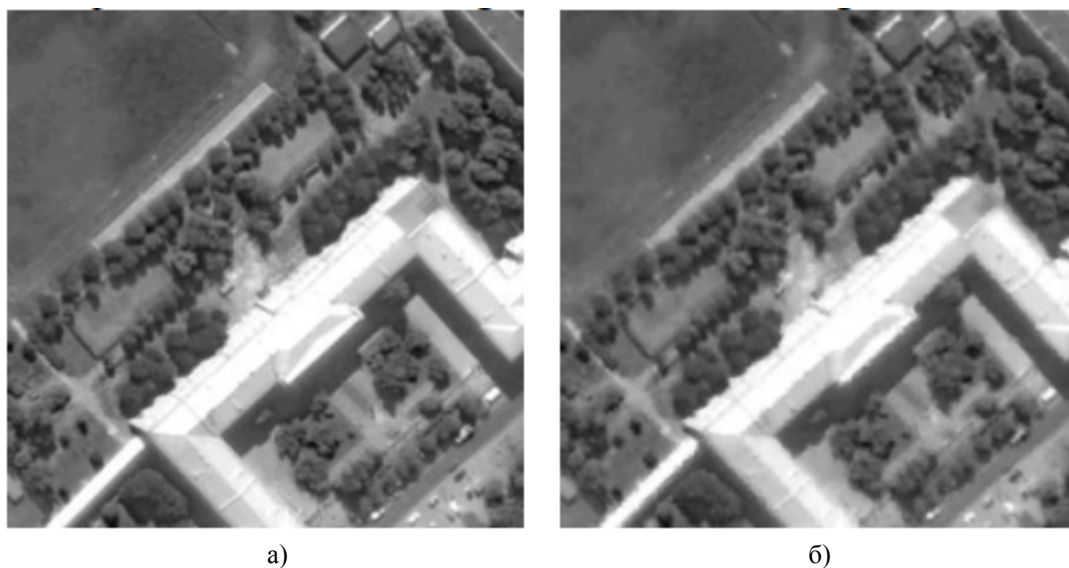


Рис. 7. Исходное изображение (а) и результат восстановления (б) после потери 80% исходных данных (1,6 бит/пикс, при СКО = 6,2)

Библиографический список

1. Saknovich L.A. Interpolation Theory and Its Applications, Germany, 1997.
2. Tarasov A., Potapova V., Svirina A. Mobile geoinformation system development 6th Mediterranean Conference on Embedded Computing. Montenegro, 2016
3. Kormen T.H. Introduction to Algorithms Text. 2th ed. Moscow, Williams, 2005, 1296 p. (Russ. ed. Krasinikov I.V., Kormen T.H.)
4. Albahari J., Albahari B. C# 5.0 in a Nutshell, Sebastopol, USA, 2012. pp 282-320.
5. Костров Б.В., Гринченко Н.Н., Геращенко Е.С., Потапова В.Ю., Тарасов А.С., Токарев А.В. Моделирование канала передачи аэрокосмических изображений, на основе каскадных кодов. Известия ТулГУ, Технические науки, Тула, 2016, 124-129 сс.

УДК 004.832; ГРНТИ 28.23.37

ПРИМЕНЕНИЕ ПРОГРАММНО-АППАРАТНЫХ СРЕДСТВ ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ТРАТ КЛИЕНТОВ

А.А. Суслина, А.Т. Кежватова

Рязанский государственный радиотехнический университет,

Российская федерация, Рязань, suslina69@mail.ru

Аннотация. В данной статье описано, как по приобретаемым клиентами товарам и услугам можно сделать вывод о сезонности покупок, почему поставщикам товаров и услуг важно знать о том, что будет пользоваться спросом у покупателей. Также иллюстрируется применение программы Deductor для проведения анализа и методы прогнозирования трат клиентов на предстоящий промежуток времени с помощью градиентных алгоритмов.

Ключевые слова. Транзакция, градиентные алгоритмы, метод градиентного спуска, Deductor.

APPLICATION OF SOFTWARE FACILITIES FOR ANALYSIS AND FORECASTING COSTS OF CUSTOMERS

A.A. Suslina, A.T. Kezhvatova

The Ryazan State Radio Engineering University

Russian Federation, Ryazan, suslina69@mail.ru

Abstract. This article describes how acquired by customers products and services can be concluded shopping season, why the suppliers of goods and services is important to know what will be in demand among buyers. The use of the Deductor program for analysis is also illustrated, and the forecasting of customer costs for the coming period of time using gradient algorithms is considered.

Keywords. Transaction, gradient algorithms, gradient descent method, Deductor.

Введение

Анализ трат клиентов с учетом сезонности и типа магазинов, в которых были совершены транзакции, является достаточно актуальной и, в то же время, малоизученной задачей. Для того чтобы провести оценку потребности клиентов и выявить их спрос на различные виды товаров и услуг требуется проанализировать траты клиентов, учитывая сезонность и тип магазина, в котором была проведена транзакция. Результаты данного анализа являются источником ценной статистики как для банка, с помощью услуг которого были совершены покупки, так и для поставщиков товаров и услуг.

Важность проведения детального анализа трат клиентов с учетом сезонности и типа магазинов для владельцев торговых точек заключается в распределении магазинов в той местности и в тот период, где и когда спрос на товар, представленный в данных магазинах, оказывается наиболее высоким. Для банка, с помощью услуг которого были совершены транзакции, важность данной задачи заключается в возможности нахождения индивидуального подхода к каждому из своих клиентов и в предоставлении ему наиболее подходящего спектра финансовых услуг.

Источником данных для предлагаемого анализа является коллекция структурированной информации, которую ПАО «Сбербанк» предоставлял для проведения конкурса Sberbank Data Contest в 2016 году. Данная выборка содержит более 1 000 000 обезличенных данных по транзакциям клиентов по 186 категориям товаров.

Проведение анализа трат клиентов

Для проведения анализа были взяты данные по транзакциям клиентов банка за две недели. Чтобы сделать вывод, какие товары пользовались спросом у клиентов в течение этих двух недель, была использована программа Deductor. Сначала данные были сгруппированы по, так называемому, тсс коду, который характеризует категорию товаров или услуг, а затем

найдена сумма транзакций по каждой категории. Результаты, полученные в ходе исследования, представлены на рисунке 1.

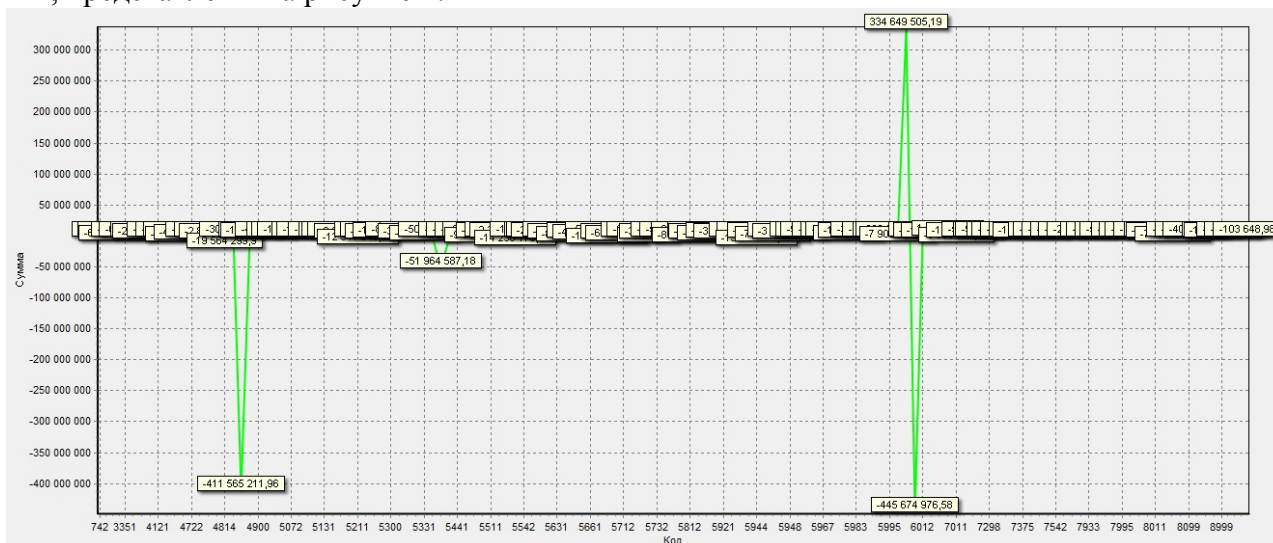


Рис. 1. Сумма транзакций по категориям товаров

Как видно из представленного на рисунке 1 графика, выделяются траты по трем категориям, имеющим следующий тсс код: 4829 (денежные переводы) – 411 565 211 рублей, 6011 (снятие наличности автоматически) – 445 674 976 рублей и 6010 (снятие наличности вручную) – 334 649 505 рублей. Также можно выделить еще одну категорию – 5411 (бакалейные магазины и супермаркеты), где покупатели потратили 51 964 587 рублей. По этим данным сложно говорить о какой-то сезонности или закономерности в приобретении и использовании товарами и услуги, поэтому необходимо провести более детальный анализ транзакций.

Для анализа рассмотрим транзакции, размер которых превышает 7 миллионов рублей. Данные приведены в таблице 1.

По данным таблицы 1 можно сделать предположение о сезонности: вероятнее всего эти транзакции характеризуют весенне-летний период отпусков, когда клиенты приобретают авиабилеты, бронируют отели, посещают рестораны и кафе и покупают фототехнику и товары для летнего отдыха, такие как плавательные бассейны. О верности предположения о сезонности также можно судить и по приобретению строительного оборудования и техники. Повышенная заинтересованность к ювелирным магазинам, магазинам часов и одежды может говорить о скором наступлении какого-то праздника или торжества.

Для более полного анализа рынка товаров и услуг и проведения соответствующих мероприятий по оптимизации деятельности делается прогноз о том, что вероятнее всего будут приобретать клиенты в предстоящий промежуток времени.

Задачу прогнозирования можно решить с применением скользящей средней, которая используется для сглаживания временных рядов, авторегрессионной модели, многомерной регрессии, метода наименьших квадратов. Еще один метод решения - это применение аппарата нейронных сетей, которые широко распространены в настоящее время. Нейронная сеть имеет ряд преимуществ, главным из которых является способность сети обучаться, что позволяет не строить четкого алгоритма для анализа. Также сеть показывает хорошие результаты в работе с нечеткими и зашумленными данными и предоставляет достаточно точный прогноз.

Таблица 1. Транзакции для анализа и их сумма

мсс код транзакции	Наименование транзакции	Сумма транзакции
4511	Авиаинии, авиакомпаний, нигде более не классифицированные;	7 479 087
4814	Звонки с использованием телефонов, считывающих магнитную ленту;	19 564 299
5211	Лесо- и строительный материал;	12 941 349
5499	Различные продовольственные магазины — рынки, магазины со спец-ассортиментом, продажа полуфабрикатов, фирменных блюд, продажа с помощью торговых автоматов;	9 343 772
5651	Одежда для всей семьи;	8 191 793
5691	Магазины мужской и женской одежды;	11 018 283
5722	Бытовое оборудование;	9 726 336
5812	Места общественного питания, рестораны;	8 973 738
5944	Магазины по продаже часов, ювелирных изделий и изделий из серебра;	13 639 623
5946	Магазины фотооборудования и фотоприборов;	7 913 566
5999	Плавательные бассейны — распродажа;	7 900 443
7011	Отели, мотели, базы отдыха, сервисы бронирования;	9 359 319
8043	Оптика, оптические товары и очки.	7 853 529

Важной частью построения нейронной сети является выбор метода ее обучения. Для решения задачи регрессии обычно применяются методы градиентного спуска, по которым можно предсказать значение y_i , зная значение x_i . На этапе обучения нейронной сети происходит минимизация ошибки (в задачах регрессии это среднеквадратическая ошибка (MSE)), которая заключается в подборе соответствующих параметров. Величина ошибки показывает, насколько точно обученная модель предсказывает целевую переменную по входным данным [2].

Рассмотрим математическую составляющую метода градиентного спуска, который может быть применен к решению поставленной задачи.

Метод градиентного спуска

Под методом градиентного спуска понимается нахождение экстремума (максимума или минимума) функции путем движения вдоль градиента. Градиентом называется то направление, в котором функция возрастает, антиградиентом — то направление, в котором функция убывает. Чтобы определить направление необходимо взять производную от функции, если это функция с одной переменной, или частные производные, если функция с несколькими переменными [1].

Разберем метод градиентного спуска на следующем примере.

Возьмем функцию f , зависящую от трех переменных x, y, z . Так как были взяты три переменные — вычислим частные производные $\frac{df}{dx}$, $\frac{df}{dy}$, $\frac{df}{dz}$. После того, как производные будут вычислены, необходимо образовать вектор, представляющий собой градиент функции:

$$\text{grad } f(x, y, z) = \frac{df(x, y, z)}{dx} * \vec{i} + \frac{df(x, y, z)}{dy} * \vec{j} + \frac{df(x, y, z)}{dz} * \vec{k},$$

где $\vec{i}, \vec{j}, \vec{k}$ – единичные векторы, параллельные осям координат [3].

Для того, чтобы охарактеризовать по каждой независимой переменной в отдельности изменение функции f требуется частные производные, а вектор градиента, который мы получили с помощью частных производных, помогает определить поведение функции в окрестности точки (x, y, z) и определить направление, в котором возрастает функция в данной точке, и направление, в котором функция в данной точке убывает (антиградиент).

Если взять модуль от градиента, то можно определить скорость, с которой функция возрастает в направлении градиента или убывает в направлении антиградиента. Для других направлений скорость изменения функции в точке (x, y, z) будет меньше значения модуля градиента:

$$|\text{grad } f(x, y, z)| = \sqrt{\left(\frac{df}{dx}(x, y, z)\right)^2 + \left(\frac{df}{dy}(x, y, z)\right)^2 + \left(\frac{df}{dz}(x, y, z)\right)^2}.$$

Идея метода состоит в движении в направлении наиболее быстрого убывания функции, то есть в движении к минимуму. Реализация идеи происходит следующим образом.

Выберем любым из способов начальную точку, в которой вычислим градиент функции. Затем, сделав небольшой шаг в обратном градиенту направлении (то есть в направлении антиградиента) в другой точке, в которой значение функции будет меньше, чем было первоначально. Процесс движения в направлении антиградиента будет продолжаться, и будет происходить движение в сторону убывания функции.

Метод градиентного спуска требует вычисления градиента целевой функции на каждом шаге. Если функция задана аналитически, то вычислить градиент на каждом шаге не составит труда, так как для частных производных, которые определяют градиент, явные формулы получить можно. В противном случае, частные производные в точках придется вычислять приближенно, производя их замену на разностные отношения [3]:

$$\frac{df}{dx} \sim \frac{f(x_1, \dots, x_i + \Delta x_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{\Delta x_i}.$$

При таких расчетах Δx_i нельзя брать слишком малым, а значения функции нужно вычислять с высокой степенью точности, иначе при вычислении разности будет допущена большая ошибка:

$$\Delta f(x_1, \dots, x_i + \Delta x_i, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n).$$

Вывод

Применяя современные программные средства и технологии в сфере IT, можно проводить анализ рынка товаров и услуг, что позволит понять поведение клиентов и предоставить в предстоящем периоде то, что их в большей степени заинтересует. Таким образом, оптимизация процессов предоставления услуг и продажи товаров потребителям может найти поддержку в интеллектуальном анализе сезонных потребительских предпочтений клиентов, основанном на применении машинного обучения.

Библиографический список

1. Афанасьев, В.Н. Анализ временных рядов и прогнозирование: Учебник / В.Н. Афанасьев, М.М. Юзбашев. - М.: ФиС, ИНФРА-М, 2012. - 320 с.
2. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. - М.: ДМК Пресс, 2015. - 400 с.
3. Гасников А.В. Современные численные методы оптимизации. Метод универсального градиентного спуска. М.: МФТИ, 2018. - 98 с.

УДК 681.3.06

**СНИЖЕНИЕ ИЗБЫТОЧНОСТИ ДАННЫХ В РЕЛЯЦИОННЫХ
БАЗАХ ДАННЫХ ЗА СЧЕТ ПРОВЕРКИ ЕЕ СТРУКТУРЫ
МЕТОДОМ ТАБЛО****А.И. Баранчиков, Н.З. Нгуен***Рязанский государственный радиотехнический университет
Россия, Рязань, nguyennngocduong1991@gmail.com*

Аннотация. В этой статье рассмотрена проблема избыточности данных в базах данных – это нежелательное явление, поскольку ведет к увеличению объема хранимой информации и при работе с этими базами данных возникают проблемы связанные с обработкой информации. Также предложен алгоритм, использующий табло, для проверки логической структуры базы данных, с его помощью выводится правильно построенная база данных, следовательно, избыточность в базах данных устраняется.

Ключевые слова. База данных, снижение избыточности, алгоритм, логическая структура, табло

**DATA REDUNDANCY REDUCTION IN RELATIONAL DATABASE
BY CHECKING ITS STRUCTURE WITH A TABLEAU****A.I. Baranchikov, N.D. Nguyen***Ryazan State Radio Engineering University
Russia, Ryazan, nguyennngocduong1991@gmail.com*

Abstract. The paper has considered the issue of data redundancy in databases – this is an undesirable phenomenon because it leads to an increase in the amount of stored information, and when working with these databases, problems of information processing arise. Also, an algorithm using tableaux to checking the database logical structure is proposed. With this algorithm, a properly constructed database is given, hence, the redundancy in databases is eliminated.

Keywords. Database, redundancy reduction, algorithm, logical structure, tableau

Введение

При работе с базами данных, для разработчиков одной из важных проблем является избыточность информации. Избыточность означает превышение информации. Избыточность вызывает, прежде всего, дублирование данных, это приводит к увеличению требуемого объема памяти, используемой для хранения данных. Самой большой проблемой при обработке данных с избыточностью является аномалии. Аномалии появляется при выполнении операции вставки, удаления и модификации данных из базы данных. Поскольку аномалии приводят к противоречиям в БД либо существенно усложняет обработку данных. Для уменьшения избыточности применяется сжатие данных без потерь. Если база данных имеет корректную логическую структуру, то вышесказанные проблемы будут устраняться. Базы данных без избыточности позволяет ускорить скорость обработки данных и обеспечить целостность данных в базе данных. Поэтому построение алгоритма для проверки структуры базы данных на правильность построения является необходимым для решения этой проблемы [1,2,4,5].

Целью данной статьи является разработка алгоритма проверки логической структуры базы данных для снижения избыточности данных, позволяющая улучшение качества баз данных и сокращение времени ее обработки.

Разработка алгоритма для проверки логической структуры базы данных

С помощью методом декомпозиции отношения, не использующим табло, и поиска потерь также можно проверить логическую структуру баз данных. Процесс декомпозиции требуется иметь всю информацию схемы баз данных, т.е. все атрибуты и связи между ними, и зависимости, имеющиеся в ней. Но из-за того, что отсутствие структуризации при хранении данных, процесс обработки очень трудоемкий.

Разработанный алгоритм проверки логической структуры схем основан на процессе декомпозиции схемы и использовании табло для определения потерь данных. В результате выполнения алгоритма будет дан ответ на вопрос: возможна ли декомпозиция без потерь отношения из множества ограничений C в R .

Табло называется табличным методом. Табло применяется для представления *проекция-соединение* отображения (*PJ-отображений*). Множество атрибутов, именуемых столбцы табло (в данной статье $A B C D$), образует схему табло. Кортежам отношения соответствует строки табло. С помощью табло, полная информация баз данных можно храниться в визуализированном и удобном для обработки виде без использования никаких дополнительных данных или параметров. [3,8]

Редукция табло это табло, которое состоит из множества всех строк исходного табло, таких, что ни одна из них не поглощается другими строками. Если редукция двух табло тождественны, то тогда два табло называется эквивалентным [2,3,6].

Данный алгоритм применяет метод прогонки (*chase*). Это вычислительный метод, с помощью которого для заданного табло T и множества зависимостей C строится новое табло T^* , такое, что $T \equiv T^*$, и T^* как отношение принадлежит $SAT(C)$, который является подмножеством множества схем базы данных, удовлетворяющий C . С помощью алгоритма прогонки табло проверяется на эквивалентность по C . [3,7].

Для успешной проверки выводимости зависимостей соединения необходимо выполнение условия $TR \equiv cTI$, где TI – табло, состоящее из одной строки выделенных переменных. Табло TI является тождественным отображением для всех отношений. Буква c означает существование множества всевозможных ограничений, применимых к табло. Эквивалентность $T1 \equiv cT2$ можно проверить с условием:

$$chase\ c(T1) \equiv chase\ c(T2),$$

т.е. когда финальное табло $T1$ по алгоритму прогонки *chase* эквивалентно финальному табло $T2$.

То есть, достаточно выполняться следующие условия

$$chase\ c(TR) \equiv chase\ c(TI).$$

поскольку $chase\ c(TI) = TI$, то имеем

$$chase\ c(TR) \equiv TI.$$

Отсюда необходимым и достаточным условием для проверки является присутствие строки выделенных элементов в $chase\ c(TR)$.

Аналогичным образом, для успешной проверки выводимости функциональных зависимостей вида $X \rightarrow Y$ необходимым и достаточным условием проверки является наличие только выделенных элементов в столбце, соответствующем атрибуту Y , финального табло по методу прогонки $chase\ c(TR)$.

Изложим алгоритм проверки:

Входные данные: схема базы данных R и множество ограничений C .

Общая схема данного алгоритма представлена на рисунке 1:

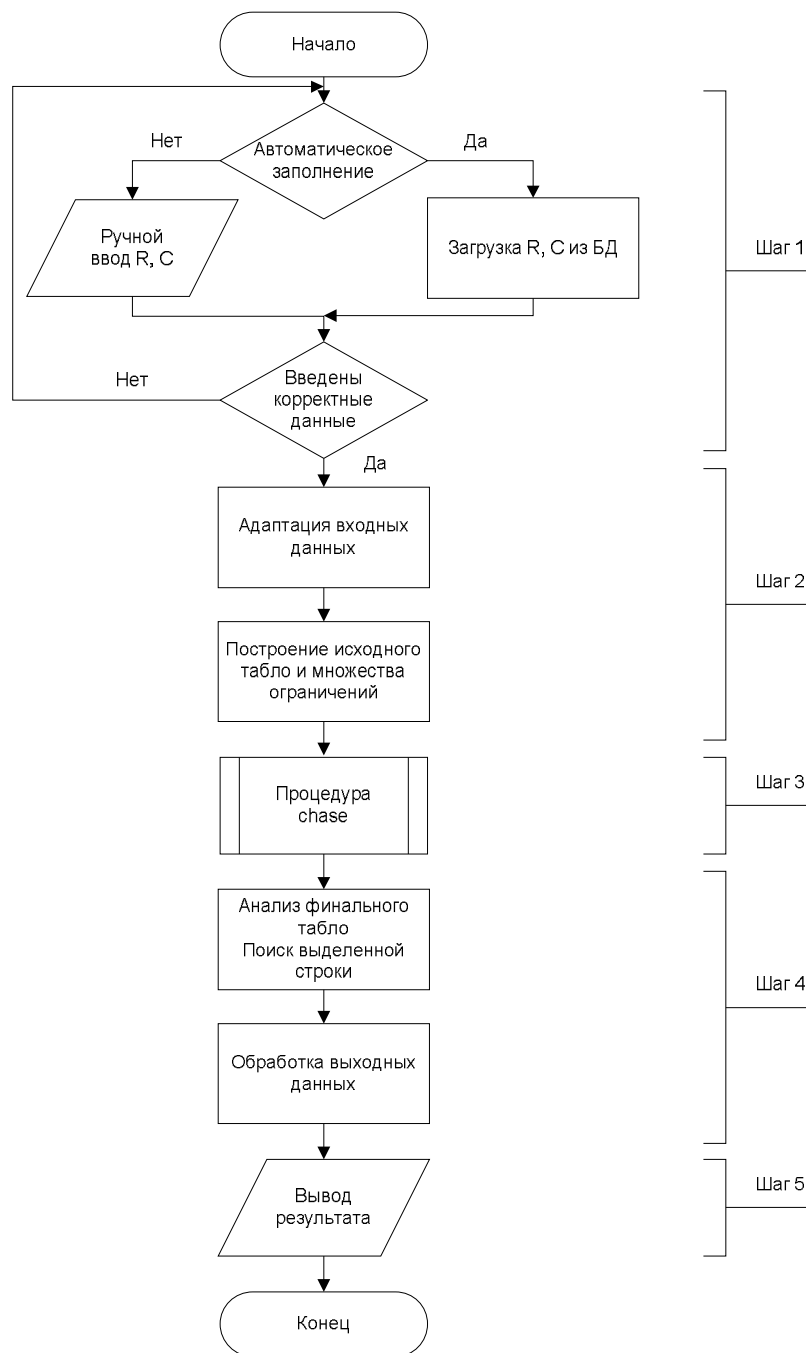


Рис. 1. Общая схема алгоритма

Рассмотрим работу алгоритма по шагам более подробно.

На первом шаге имеется правильный ввод входных данных: схемы R и множества ограничений C , которые являются совокупностью F - и J -правил, т.е. правила преобразования для множество ограничений C , составленных из F - и J -зависимостей. Эти правила позволяет проверять правильность введенных данных.

На втором шаге алгоритма выполняется адаптация входных данных, т.е. данные преобразуется таким образом, что их можно удобно для обрабатывать, и построение исходного табло TR на схеме R .

На третьем шаге - алгоритм прогонки выполняется.

Вычисление по методу прогонки описываются довольно просто. А именно: для

заданных T и C применяются F - и J -правила, соответствующие F - и J -зависимостям из C , до тех пор, пока изменяется в T и C . При этом порядок применения правил преобразования несуществен.

На четвертом шаге оценивается эквивалентность. Для J -зависимостей финальное табло T^* проверяется на эквивалентность с TI (табло, состоящим из одной строки выделенных элементов). Если в T^* существует строка выделенных элементов, то $T^* \equiv TI$. Для F -зависимостей проверяется наличие в соответствующем столбце только выделенных элементов.

На пятом шаге выводится результат.

Метод прогонки и проверка правильности показаны на рисунке 2.

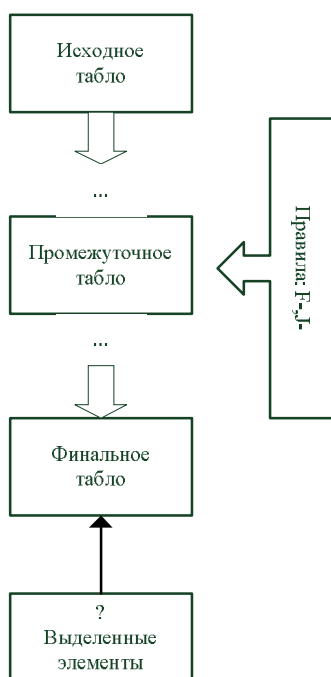


Рис. 2. Метод прогонки и проверка правильности

Пример работы разработанного алгоритма

Пусть дана схема базы данных $R=\{AB, BC, AD\}$ и множество ограничений $C=(A \rightarrow D, *[AB, BCD])$. Применим разработанный алгоритм для получения финального табло. Исходное $T1$ и финальное $T1^*$ табло представлены на рисунке 3.

T1					T1*				
(A	B	C	D)		(A	B	C	D)	
a1	a2	b1	b2		a1	a2	b1	a4	
b3	a2	a3	b4		b3	a2	a3	a4	
a1	b5	b6	a4		a1	b5	b6	a4	
					a1	a2	a3	a4	
					b3	a2	b1	a4	

Рис. 3. Исходное и финальное табло для R, C

Поскольку $chase\ c(T1)$ содержит строку выделенных элементов, то любое отношение из $SAT(C)$ без потерь разлагается в R .

Для схемы базы данных $S=\{AB, DC, CD\}$ $chase\ c(T2)$, найденное по разработанному алгоритму, не содержит строки выделенных элементов. В $SAT(C)$ существуют отношения,

имеющие потери при декомпозиции в S . Исходное T_2 и финальное T_2^* табло представлены на рисунке 4.

T2				
(A	B	C	D)	
a1	a2	b1	b2	
b3	a2	a3	b4	
b5	b6	a3	a4	

T2*				
(A	B	C	D)	
a1	a2	b1	b2	
b3	a2	a3	b2	
b5	b6	a3	a4	
a1	a2	a3	b2	
b3	a2	b1	b2	

Рис. 4. Исходное и финальное табло для S , C

Выводы

В данной работе разработан алгоритм, использующий табло, для проверки структуры баз данных. Этот алгоритм является частью алгоритма коррекции структуры базы данных. С помощью этого алгоритма выводится правильно построенная база данных, следовательно, избыточность в базах данных устраняется. Это позволяет ускорить скорость обработки данных и обеспечить целостность данных в реляционной базе данных.

Библиографический список

1. К. Дж. Дейт, Хью Дарвен Основы будущих систем баз данных. Третий манифест. Перевод : С.Д.Кузнецов, Т.А.Кузнецова, Издательство Янус-К, 2004 г. - 656 с.
2. Крёнке Д. Теория и практика построения баз данных, 8-е изд. "Питер", 2003. - 800 с.
3. Мейер Д. Теория реляционных баз данных: Пер. с англ. - М.: Мир, 1987. - 608с.
4. Малихина М. П. Базы данных. Основы, проектирование, использование. – СПб.: БХВ-Петербург, 2004. – 512 с.
5. Ахо Альфред В., Хопкрофт Джон Э., Ульман Джеффри Д. Структуры данных и алгоритмы: Пер. с англ. – М.: Вильямс, 2000. – 384 с., ил.
6. Хомоненко А. Базы данных. –Бином-Пресс, 2007.–736 с
7. Дейт К. Руководство по реляционной системе DB2 // М.: Финансы и статистика. 1988. 320 с
8. Баранчиков А.И., Дрожжин И.В., Громов А.Ю. Алгоритм поиска ошибок и избыточности в структуре баз данных // Известия ТулГУ. Технические науки. 2013

УДК 519.872; ГРНТИ 50.05.19

ВИРТУАЛЬНАЯ СРЕДА МОДЕЛИРОВАНИЯ EVE-NG

Н. Морозов

Рязанский государственный радиотехнический университет

Россия, Рязань, moorozoov@yandex.ru

Аннотация. В данной статье рассказывается о виртуальной среде моделирования сетей EVE-NG и её применении в учебном процессе. Проводится сравнительный анализ и ключевые преимущества использования данной среды.

VIRTUAL MODELING ENVIRONMENT EVE-NG

N. Morozov

Ryazan State Radioengineering University

Russia, Ryazan, moorozoov@yandex.ru

Abstract. This article deals with virtual environment for modeling that named EVE-NG and it's usage in the process of studying. A comparative analysis and key features of it's usage are carried out.

В современном мире в процессе обучения целесообразно применять различные новые информационные технологии, которые позволяют сделать процесс обучения более удобным и интерактивным. На данный момент для изучения сетевых технологий в подавляющем

большинстве случаев используется *Cisco Packet Tracer*. Данный программный комплекс предоставляет пользователю широкий спектр возможностей и достаточно большое количество различного оборудования фирмы *Cisco*. Однако, он имеет ряд существенных недостатков. Во-первых, реализованная функциональность устройств ограничена и не предоставляет всех возможностей реального оборудования. Во-вторых, *Packet Tracer* доступен бесплатно только для участников Программы Сетевой Академии *Cisco*. Кроме всего прочего, данный программный комплекс не поддерживает работу со сторонним оборудованием вследствие отсутствия возможности загрузки и добавления образов сетевых устройств.

Все приведенные выше недостатки полностью устранены в виртуальной лаборатории *EVE-NG (Emulated Virtual Environment Next Generation)* – обновленной версии *UNetLab*. *EVE-NG* не только наследует весь уже имеющийся функционал *UNetLab*, но и заметно расширяет его, добавляя удобство и простоту работы. Основными отличительными чертами и достоинствами *EVE-NG* являются:

- поддержка виртуализации широкого спектра сетевого оборудования различных производителей: *Cisco*, *Juniper*, *Alcatel*, *Dell*, *Aruba* и т.д.;
- наличие возможности запуска виртуальных хостов с популярными операционными системами и организация взаимодействия между ними;
- многопользовательский режим с распределением прав доступа к созданным и используемым лабораторным стендам;
- удобный веб-интерфейс, возможность дистанционного доступа;
- *EVE-NG* может быть установлена как виртуальная машина под *VMware Workstation* или непосредственно на компьютер с ОС *Ubuntu Linux*;
- лабораторные стенды легко перемещаются между различными VM;
- присутствует уникальная возможность создавать наглядные топологии на базе изображений в формате *.png*, которые преобразуются в интерактивные структуры.

Для работы *EVE-NG* потребуется:

- процессор Intel (уровня Xeon) с полной поддержкой VT-x/EPT и не менее 16 гигабайтов оперативной памяти;
- для установки и работы с серверной частью потребуются базовые навыки администрирования Linux;
- понимание принципов работы FTP, SSH, Telnet, VNC – это потребуется для управления различным оборудованием;
- базовый технический английский язык.

EVE-NG постоянно обновляется, расширяя список поддерживаемого оборудования. В настоящий момент актуальной является версия 2.0.3-86. Она поддерживает эмуляцию *Dynamips*, *IOL* и *Qemu*. *Dynamips* применяется для эмуляции *Cisco IOS*, *IOL* – *Cisco IOS on Linux*, *Qemu* – эмуляция операционных систем, ASA, оборудования других производителей.

Для *EVE-NG* был создан курс методических указаний к лабораторным работам. Курс имеет следующую структуру:

1. Установка и ознакомление с *EVE-NG*. В данной работе приводится подробная инструкция по установке *EVE-NG* на *VMWare Workstation*, приводится пример загрузки образов оборудования для работы в эмуляторе, описывается процесс построения сетевой топологии в эмуляторе и даётся задание для самостоятельной работы.
2. Протоколы маршрутизации. В данной работе приводятся теоретические сведения о различных протоколах маршрутизации и описывается процесс настройки маршрутизаторов на топологии для примера, после чего даётся практическое задание.

3. VLAN. В данной работе даются подробные теоретические сведения о виртуальных локальных сетях, описывается процесс создания VLAN на оборудовании Cisco и далее приводится задание для самостоятельной работы. Все основные команды для настройки оборудования также приведены в методических указаниях.
4. NAT. В этой лабораторной работе предлагается ознакомиться с механизмом преобразования сетевых адресов и после изучения теоретического материала и разбора примера применить полученные знания на практике.
5. VPN. Теоретическая часть данной работы содержит подробные сведения о группе технологий, позволяющих обеспечить одно или несколько сетевых соединений поверх другой сети. Приведен пример настройки оборудования и дано практическое задание для отработки полученных навыков.
6. Протокол BGP. Эта работа предназначена для ознакомления с одним из основных протоколов динамической маршрутизации, используемых в Интернете. Даются подробные теоретические сведения, разобран пример настройки оборудования с созданием конфедераций. В конце работы предлагается выполнить практическое задание по изученному материалу.
7. Межсетевые экраны. Изложенный в теоретической части материал касается основных вопросов, связанных с фильтрацией трафика. Разобран пример настройки оборудования, максимально приближенный к реальной ситуации. В конце работы предлагается выполнить практическое задание по настройке межсетевого экрана.
8. Образы систем. Данная лабораторная работа предназначена для изучения процесса загрузки в эмулятор EVE-NG образов операционных систем и получения навыков работы внутри них. Разобран пример загрузки и установки образа. Для закрепления навыков даётся практическое задание.

Курс работ позволяет изучить основные сетевые технологии, применяемые в повседневной жизни, на достаточном для администрирования небольшой сети уровне. Порядок работ организован таким образом, что в каждой последующей работе применяются знания из предыдущей. Такая система позволяет закрепить базовые умения и хорошо запомнить порядок действий для стандартной настройки оборудования, которое применяется достаточно широко. На рисунке 1 приведен пример топологии, которая может быть реализована средствами EVE-NG при выполнении лабораторных работ.

Для более детального изучения сетевых технологий существует достаточно большое количество обучающих ресурсов, на некоторых из которых имеется и база с практическими заданиями. Благодаря тому, что EVE-NG поддерживает широкий спектр производителей оборудования, большинство из доступных заданий будет возможно выполнить именно на том оборудовании, для которого изначально создавалось задание.

Стоит отметить, что в данной среде имеется возможность проектирования реальных моделей сетей, с учетом мощностей серверов и компьютеров пользователей, установленных межсетевых экранов, скоростей соединения, пропускных способностей портов и оборудования. Именно благодаря этому данная виртуальная среда зарекомендовала себя среди людей, занимающихся сетевыми технологиями как на базовом, так и на профессиональном уровне.

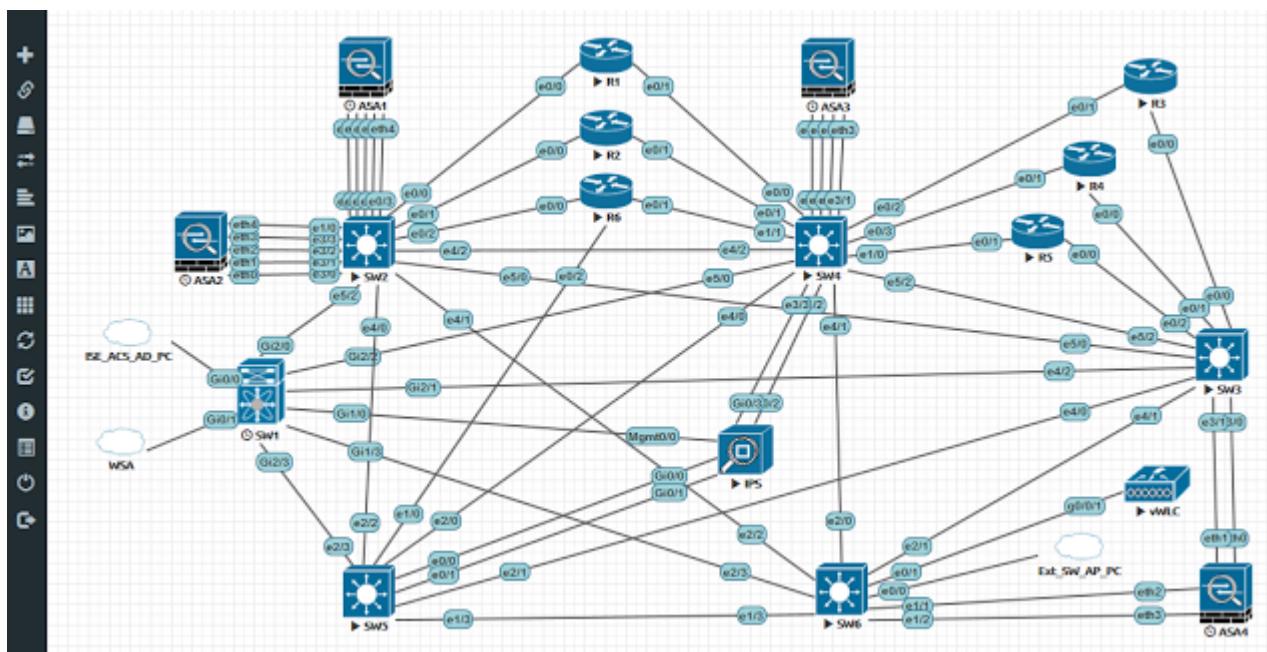


Рис. 1. Пример топологии в интерфейсе EVE-NG

Библиографический список

1. В. Олифер, Н. Олифер «Компьютерные сети. Принципы, технологии, протоколы. Учебник» 2016 г.
2. Д. Куроуз, Т. Росс «Компьютерные сети. Настольная книга системного администратора» 2016 г.
3. У. Одом «Официальное руководство Cisco по подготовке к сертификационным экзаменам CCNA ICND2 200-101. Маршрутизация и коммутация» 2016 г.

УДК 681.5 ГРНТИ 50.43.19

ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ СИСТЕМ ВИЗУАЛИЗАЦИИ ИЗОБРАЖЕНИЙ ДЛЯ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

А. Киселев, М. Никифоров

Рязанский государственный радиотехнический университет,
Россия, Рязань

Аннотация. В данной работе описаны основные этапы проектирования, разработки и сертификации оборудования, используемые для внедрения в системы воздушных судов.

Ключевые слова. Бортовая аппаратура, сертификация.

EVALUATION OF ECONOMIC EFFICIENCY OF VISUALIZATION SYSTEMS FOR AIRCRAFT APPARATUS SYSTEMS

A. Kiselev, M. Nikiforov

Ryazan State Radio Engineering University,
Russia, Ryazan

Abstract. This work describes the main stages of design, development and certification of equipment used for deployment in aircraft systems.

Keywords. Airborne equipment, certification.

Введение

Основной целью систем безопасности систем визуализации летательных аппаратов прогнозирование, оценка и анализ технических рисков в общем процессе оценки стоимости жизненного цикла авиационного оборудования. Экономическая эффективность выражается в сокращении сроков разработки новых образцов и обеспечении максимальной коммерческой

эффективности при обусловленном выполнении требований безопасной эксплуатации оборудования.

В данной статье будут рассмотрены этапы и методы разработки оборудования на предмет возможности сокращения итераций в разработке систем летательных аппаратов.

Этапы разработки оборудования

В проектной деятельности присутствуют следующие этапы оценки надежности:

1. Выявление целей в сфере надежности
2. Анализ актуальности и значимости работ в сфере надежности
3. Создание плана действий для достижения целей в сфере надежности
4. Выполнение соответствующих действий в сфере надежности
5. Анализ результатов осуществленных действий
6. Оценка достигнутых результатов для дальнейших действий по усовершенствованию

В текущий момент продвижение менеджмента качества характеризуется переносом основных акцентов с оценки готовой продукции на оценку и менеджмент процессов ее создания.

Анализ зарубежного опыта предоставил следующую информацию[2]:

1. Сертификация является ключевым средством гарантии качества
2. Методы управления качеством рациональны только тогда, когда они осуществляются на всех этапах разработки

Основным показателем, обеспечивающим гарантию проектирования, является системный подход к проектированию бортовой аппаратуры воздушных судов (ВС). Под гарантией проектирования понимается процесс, состоящий из специально запланированных систематических мероприятий, обеспечивающих в совокупности уверенность в том, что ошибки или упущения в требованиях или проекте выявлены и устранены таким образом, что реализованная система будет удовлетворять сертификационным требованиям.

При решении перечисленных задач математическое моделирование используется, когда известно достаточно достоверное математическое описание моделируемого процесса. Полу натурное моделирование применяется для оценки аппаратурных решений, эргономической оценки и при отработке образцов для уточнения технических решений, получения объективных оценок для принятия решений о проведении натурных испытаний и возможности пере хода от одного этапа испытаний к последующему.

Значительный интерес представляет задача обоснования достаточности обработанности системы на различных этапах ее разработки и возможности перехода на следующий в технологической цепочке этап.

Эта задача решается с использованием так называемых «кривых роста надежности».

На каждом этапе разработки системы происходит обнаружение и устранение различных дефектов, т. е. корректировка и доработка системы. При этом наблюдается улучшение характеристик системы. Определив отказ как любое несоответствие параметров и характеристик системы предъявляемым требованиям, и учитывая вероятностный характер процессов обнаружения дефектов, получим возможность использования для описания этого процесса хорошо известного и достаточно полно разработанного математического аппарата теории надежности.

Аналогом отказа, в данном случае, является обнаружение дефекта. Вероятность события при котором система будет работать без отказов называют надежностью системы.

При тестировании системы ее безотказность растет. Данную закономерность можно описать экспоненциально, отталкиваясь от вышеупомянутого определения надежности системы (R)[4].

$R_i(t_i) = a_i - (a_i - R_{oi}) \exp\{-Q_i t_i\}$, где этап разработки - R_i ;

t_i – разработка на данном этапе во временном выражении;

Ro_i – нулевое значение надежности (начальное);

Q_i – интенсивность обнаружения, а так же исправления дефектов на данном этапе;

a_i – маргинальное (предельное) значение надежности для данного этапа, определяемое цельностью имитации условий эксплуатации системы.

Использование экспоненциальной модели обусловлено тем, что рост надежности систем пропорционален обнаруженной ненадежности, т. е.

$$R = Q(a - R),$$

а отличие параметра «а» от единицы определяет проявление функционирования любой системы случайных отказов, не требующих переработки системы.

При исследовании динамики роста надежности различных летательных систем в процессе их экспериментальной обработки было выявлены следующие закономерности:

– интенсивность обнаружения дефектов уменьшается с переходом на следующие этапы, т. е.

$$Q_{i-1} > Q_i,$$

что объясняется как уменьшением числа неопределенных дефектов, так и возрастанием сложности в условиях их обнаружения;

– маргинальное (предельное) значение растет с переходом на следующий этап, т. е. $a_{i-1} < a_i$,

что объясняется все наиболее полной имитацией реальных эксплуатационных условий при переходе от математического моделирования к летным испытаниям.

В результате может быть поставлена и решена задача оптимального по времени поэтапного перехода, а именно:

$$T = t_{i-1} + t_i = \frac{1}{Q_{i-1}} \ln \frac{a_{i-1} - Ro_{i-1}}{a_{i-1} - Ro_i} + \frac{1}{Q_i} \ln \frac{a_i - Ro_i}{a_i - Ro_{i+1}} = \min R_{oi}$$

Решением для этой оптимизационной задачи состоит условие равенства возрастания надежности на $i-1$ -ом и i -ом этапах отработки в точке перехода Ro_i , а именно

$$Q_i(a_i - Ro_i) = Q_{i-1}(a_{i-1} - Ro_i).$$

Если изменить время на цену отработки на разных этапах, то получится аналогичное условие.

Анализируя зарубежные технологии, можно заметить, что важнейшую роль в отечественной оценке определения соответствия главных показателей летной пригодности самолетов играют летные испытания, в то время как зарубежом около 80% всех проблемных ситуаций, связанных с разработкой авиационного оборудования, решаются на земле[3].

Каждая система может разрабатываться сразу на нескольких этапах проектирования. Процесс разработки компонентов системы может включать в себя несколько жизненных циклов.

Концептуальное проектирование является первым этапом в типовом подходе к проектированию систем ВС. Обычным для данной системы является концепция «сверху вниз». В начале данного процесса проектирования находятся требуемые функции для ВС. Заканчивается проектное проектирование сертификацией.

Каждый элемент (компонент) системы создается с учетом всех требований функциональности и безопасности по четким критериям, установленным архитектурой данной системы. Вариантов реализации архитектуры системы может быть несколько.

Сроки реализации, затраты на производство, используемые технологии и готовность промышленности могут вносить дополнительные ограничения в процесс реализации.

Распределение требований между программным обеспечением и аппаратной частью вместе с проектированием архитектуры системы являются итеративными процессами, которые тесно связаны между собой. Каждый цикл системы повышает понимание производственных требований и распределяет их на более обоснованном уровне.

Необходимым условием для осуществления процесса разработки аппаратной части является проведение трассировки и отслеживание требований безопасности. Наихудшее отказное состояние может быть использовано только в случае невозможности проведения трассировки.

Интеграция программного обеспечения и аппаратной части может производиться на компьютерных эмуляторах, опытных образцах или на специальных изделиях, созданных для летных или лабораторных испытаний. Условия определяются в зависимости от вида системы. В результате испытаний получают физическое изделие с гарантийными данными уровня разработки, а так же с настраиваемой конфигурацией.

Одно изделие последовательно интегрируется за другим до полного завершения интеграции системы.

При возникновении трудностей в прогнозировании или моделировании среды, в которой планируется использование ВС, работу могут выполняться непосредственно на ВС. Несмотря на то, что при проведении интеграции на ВС получаю достоверные и экономические эффективные результаты, в условиях модели или лаборатории можно достигнуть впечатляюще высоких результатов.

Гарантии проекта оборудования

Интегрированная система существует для гарантии установки на ВС отдельного и совместного функционирования. Задачей интеграции является доказательство того, что межсистемные требования изделия будут соответствовать стандартам, а так же будет существовать возможность определить и исправить не предусмотренные функции.

Ключевые моменты, определенные процессом доказательства гарантии:

1. Своевременное определение исполнения поддерживающих процессов;
2. Оценка безопасности, взаимодействующая с другими системами поддержки

ключевых точек проекта.

Большинство сложных и высокоинтегрированных систем использует доказательство обеспечения гарантии в течении основной части периода проектирования. Весь процесс сертификации, в данном случае, рассматривается как процесс поддержки проектирования.

Доказательство соответствия ВС, а так же его систем ТНЛГ (требованиям норм летной годности), является основной целью процесса сертификации.

Необходимость обеспечения координации действий заявителя с органом сертификации обусловлена тем, что использование методов гарантии проектирования является главным средством очевидности сертификации интегрированных и сложных систем. Данную процедуру рекомендуется осуществить как можно быстрее (до реализации проекта). Сертификационный орган оценивает полноту данных для доказательства соответствия методов оценки, которые предлагает заявитель.

Аналитическая очевидность соответствия ТНЛГ определяет процесс оценки безопасности. В него входят конкретные оценки, изменяющиеся вместе с проектированием системы и связанные с остальными процессами поддержки ее разработки. Основные процессы определения безопасности[1]:

1. Оценка функциональных рисков (ФНА) – производится рассмотрение функций ВС, а так же его систем с целью определения возможных отказов, проведение классификации опасностей и присущих им отказных состояний. Данная оценка выполняется на начальном этапе проектирования и дополняется по мере поступления новых отказных состояний или новых функций.

2. Предварительная оценка безопасности системы (PSSA) – результатом данной процедуры является установление конкретных требований безопасности системы, а так же ее составных изделий. Позднее дается первоначальное подтверждение возможности удовлетворения данных требований архитектурой системы. В процессе проектирования системы предварительная оценка безопасности может корректироваться.
3. Оценка безопасности системы (SSA) – сбор, анализ и документирование доказательств, указывающих на то, что спроектированная система удовлетворяет требованиям безопасности, указанным в раках процесса оценки функциональных опасностей (FHA) и предварительной оценки (PSSA).
4. Анализ общих причин отказов (CCA) – определение и оценка требований по функциональному и физическому разделению систем. В то же время происходит проверка выполнения данных требований.

Далее выполняется процесс валидации, к целям которого относятся проверка правильности и полноты требований.

При верификации могут использоваться отдельные методы валидации. Определение планов верификации и валидации могут обуславливать данное течение процесса.

Проведение верификации дает гарантию на удовлетворение предъявляемых требований к реализуемой системе (если та прошла верификацию).

Выводы

Анализируя зарубежные подходы и процедуры сертификации, валидации и верификации изделий авиационной техники, можно определить, что отличительной чертой зарубежных технологий проектирования является сертификационная направленность подавляющего большинства видов работ. Работы производятся по принципу сквозной сертификации, для реализации которого на этапе эскиза разрабатывается система сертификаций, охватывающая все виды работ.

Лабораторно стендовые испытания также должны включать в себя процедуры верификации, для придания им сертификационной направленности. Сквозная сертификация должна гарантировать готовность сертификационной документации более чем на 50% к моменту начала летных испытаний.

Сами летные испытания, играющие важную роль для оценки основных характеристик ВС в отечественной практике, за рубежом имеют существенно меньшее влияние на процесс проектирования. Это дает возможность получить до 80% всех данных на земле, что гарантирует собой более высокий экономический эффект от разработки.

Библиографический список

1. Елесина С.И., Никифоров М.Б. Документальное сопровождение разработки и производства радиоэлектронной аппаратуры [Учебное пособие]: Рязань, 2014.- 60с.
2. Муратов Е.Р. Методы сокращения вычислительной сложности алгоритмов совмещения разнородных изображений [Статья] Электронный журнал Cloud of Science: Рязань, 2014. - 328-336 с.
3. Новиков А.И., Муратов Е.Р., Логинов А.А., Никифоров М.Б. Сокращение вычислительной сложности совмещения изображений в авиационных системах технического зрения [Статья]: Динамика сложных систем, XXI век, №1, т.9, 2015, Москва. - 33-40с.
4. Кузнецов А.Г. «Навигация и управление летательными аппаратами». 2010 –(86) №2.

УДК 004.4; ГРНТИ 50.41.29

ПРИМЕНЕНИЕ JAVASCRIPT БИБЛИОТЕКИ REACT В ЗАДАЧЕ УВЕЛИЧЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ WEB-СТРАНИЦЫ

Д.А. Левшин, Т.А. Петровская

*Рязанский государственный радиотехнический университет,
Российская Федерация, Рязань, he.cebe@yandex.ru*

Аннотация. В данной работе рассматривается задача применения библиотеки React для повышения эффективности работы сайта, а также анализируются достоинства и недостатки её использования.

Ключевые слова. JavaScript, React, web-программирование

APPLICATION OF JAVASCRIPT REACT LIBRARIES IN THE PROBLEM OF INCREASING THE PERFORMANCE OF THE WEB-PAGE

D.A.Levshin, T.A.Petrovskaja

*The Ryazan State Radio Engineering University,
Russian Federation, Ryazan, he.cebe@yandex.ru*

Abstract. This paper, we consider the problem of using the React library to improve the efficiency of the site, as well as analyze the advantages and disadvantages of its use

Keywords. JavaScript, React, web-programming

Введение

В настоящее время в DOM (document object model) web-страниц постоянно увеличивается количество узлов. Работа с ними может осуществляться с помощью языка JavaScript и библиотек на его основе, например jQuery. Проблема этих подходов заключается в том, что они существенно замедляют работу сайта. Добиться увеличения эффективности позволяет библиотека React и её концепция Virtual DOM. React — это библиотека для создания пользовательских интерфейсов. Она представляет собой хороший инструмент для построения web-приложений. Библиотека проста в использовании и освоении за счет лаконичного и понятного синтаксиса.

Особенности библиотеки

Одной из особенностей данной библиотеки является использование языка программирования JSX (JavaScript and XML), который является сочетанием HTML и JavaScript. Высокая производительность приложения обеспечивается за счет концепции Virtual DOM. Взаимодействие осуществляется с легковесной копией обычной DOM-модели сайта. После внесения необходимых изменений в Virtual DOM, React сравнивает исходную DOM и его виртуальную копию, находит изменившиеся узлы в этой модели. Затем алгоритмы React определяют минимальное количество операций, необходимых для приведения обычной DOM модели сайта в соответствие с его виртуальной моделью. В отличие от работы с DOM — моделью на чистом JavaScript или jQuery, где при изменениях модель перерисовывается полностью, React перерисовывает только те узлы, которые подверглись изменению. За счет этого производительность сайта увеличивается [2].

Ещё одной отличительной особенностью React является построение приложения на основе компонентного подхода. Компоненты позволяют разделять интерфейс на самостоятельные части с возможностью повторного использования и анализировать каждую из них в отдельности. Они представляют собой JavaScript функции, которые принимают свойство и возвращают React элементы, которые описывают то, что должно выводиться на экране.

Применение React предоставляет ряд преимуществ для web-разработчика: во-первых, скорость работы web-приложения значительно увеличивается за счет работы с Virtual DOM, во-вторых, увеличивается скорость разработки, упрощается процесс тестирования за счет

возможности повторного использования кода, в-третьих, факт, что эта библиотека распространяется и поддерживается высококвалифицированными специалистами и становится популярнее с каждым годом, дает основания полагать, что тенденция к дальнейшим улучшениям продолжится [1].

У React есть некоторые недостатки: сложная документация (информация на официальном сайте не структурирована); не все стандартные браузеры поддерживают React (преодолеть эту проблему помогут различные плагины, но, учитывая вес библиотеки, их использование нужно минимизировать); React достаточно молод, поэтому большинство стандартных виджетов (компонент расширений, компонент интерфейса) придется описывать самостоятельно [3].

Экспериментальная часть

На рисунке 1 представлен самый простой React компонент. Он состоит из заголовка первого уровня и кнопки увеличения счетчика.

Текущее значение счетчика: 0

Увеличить на единицу

Рис.1. Начальное состояние компонента

Для начала необходимо в шапке web-страницы подключить необходимые библиотеки («react» и «react-dom») и компилятор «Babel», который переводит код, написанный на ES6, в ES5-совместимый код. Код html-страницы представлен ниже:

```
<!DOCTYPE html>
<head>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.0/react.js"
></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/react/0.14.0/react-
dom.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-
core/5.6.15/browser.js"></script>
  <title>Document</title>
</head>
<body>
  <div id="root">
    </div>
    <script type='text/babel'>
class App extends React.Component{
  constructor() {
    super();
    this.state = {
      count:0
    }
  }
  handleClick() {
```

```
        this.setState({
          count:++this.state.count
        })
      }
      render() {
        return (
          <div>
            <h1>Текущее значение счетчика:
{this.state.count}</h1>
            <button type='text'
onClick={this.handleClick.bind(this)}>
              Увеличить на единицу
            </button>
          </div>
        )
      }
    }
    ReactDOM.render(
      <App />,
      document.getElementById('root')
    )
  </script>
</body>
</html>
```

При инициализации компонента первоначальное значение счетчика равно 0. После нажатия на кнопку, вызывается функцию «handleOnClick», которая увеличивает значение счетчика на единицу и обновляет состояние компонента. После чего React перерисовывает только значение обновленного счетчика. Обновленное состояние компонента показано на рисунке 2.

Текущее значение счетчика: 1

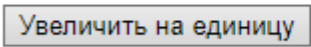


Рис. 2. Обновленное состояние компонента

На последнем этапе происходит «встраивание» созданного компонента в web-страницу, путем вызова метода «render» у объекта «ReactDOM», которому в качестве первого аргумента передаётся сам компонент, а в качестве второго аргумента – занимаемое место на web-странице.

Выводы

Работа с компонентами, легкость поддержания и повторного использования кода превращают работу с библиотекой React в непрерывный процесс улучшения. Используя Virtual DOM и изоморфный JavaScript, разработчики могут с высокой скоростью создавать производительные web-приложения, в не зависимости от их сложности. Возможность заново использовать уже написанные ранее компоненты заметно увеличивает скорость разработки и тести-

рования приложения, а это отражается на уменьшении затрат в разработке. Библиотека React является трендом на рынке современных информационных технологий, а её применение в новых проектах, является уже необходимостью.

Библиографический список

1. Бэнкс А., Порселло Е. React и Redux, Санкт-Петербург, 2018. – 45 с.
2. Стефанов С. React JS Быстрый старт, Санкт-Петербург, 2017. – 85 с
3. Федосеев А. React.JS Essentials, Москва, 2016. – 102 с.

УДК 004.421; ГРНТИ 50.51.15

ПРИМЕНЕНИЕ МЕТОДА КАРТЫ РАССТОЯНИЙ В ЗАДАЧЕ ПРОЕКТИРОВАНИЯ ДВИЖЕНИЯ БЕСПИЛОТНОГО НАЗЕМНОГО УСТРОЙСТВА

Т.А. Петровская, А.Т. Кежватова

*Рязанский государственный радиотехнический университет,
Российская федерация, Рязань, tata_trk@mail.ru*

Аннотация. В данной работе рассматривается задача проектирования движения беспилотных наземных объектов, а также анализируются достоинства и недостатки метода карты расстояний.

Ключевые слова. Карта расстояний, проектирование движения, беспилотные устройства

APPLICATION OF THE DISTANCE FIELD METHOD IN THE PROBLEM OF DESIGNING THE MOVEMENT OF THE UNBEILED GROUND DEVICE

T.A. Petrovskaia, A.T. Kezhvatova

*The Ryazan State Radio Engineering University
Russian Federation, Ryazan, tata_trk@mail.ru*

Abstract. In this paper we consider the problem of designing the motion of unmanned ground objects, and also analyze the advantages and disadvantages of the distance map method.

Keywords. Distance field, motion design, unmanned devices

Введение

Проектирование пути перемещения является одной из сложнейших задач при построении беспилотных устройств. Из всего ряда беспилотных устройств наземные транспортные средства обладают рядом существенных кинематических ограничений. При планировании движения следует уделять особое внимание не только текущему положению машины и ускорению, но и окружающим его объектам [5]. Для проектирования пути применяются различные методы, одним из которых является классический алгоритм на графах. Применение метода карты расстояний (Distance Field) позволяет найти наикратчайший путь с учетом расположения препятствий. Построение карты расстояний – метод, основанный на преобразовании исходного изображения в новое бинарное изображение, где каждый пиксель имеет значение, соответствующее минимальному расстоянию до фонового пикселя.

Алгоритм

Шаг 1. *Преобразование изображения местности со спутника в монохромное изображение*

Цветное изображение чаще всего формируется с использованием палитры RGB, состоящей из трех каналов: красного, зеленого, синего. Для преобразования снимка со спутника необходимо найти порог яркости, что возможно осуществить при помощи различных кри-

териев: Оцу, Бернсена, Эйквеля, Ниблэка. Метод Оцу ищет порог, уменьшающий дисперсию внутри класса, которая определяется как взвешенная сумма дисперсий двух классов [1]:

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t),$$

где w_i – это вероятности двух классов, разделенных порогом t ;
 σ_i^2 – дисперсия этих классов.

При известном значении порога яркости преобразование в монохромное изображение осуществляется при помощи следующей формулы [3]:

$$T(x, y) = \begin{cases} 255, & \text{если } S(x, y) > \sigma_w^2(x, y) \\ 0, & \text{в остальных случаях} \end{cases},$$

где $T(x, y)$ – конечное изображение;
 $S(x, y)$ – исходное изображение;
 $\sigma_w^2(x, y)$ – порог яркости.

Шаг 2. Построение карты расстояний

Первоначально в бинарном изображении устанавливается значение пикселя 0 или 1 в зависимости от представляемого им объекта, а именно:

$$\begin{aligned} R &= 1, \text{ если точка объекта} \\ \bar{R} &= 0, \text{ если фоновая точка,} \end{aligned}$$

где R и $\bar{R}$ – пиксели исходного изображения.

Существует два подхода к построению карты расстояний: метод 4 соседей и метод 8 соседей, где наиболее точный последний. Тогда пусть пиксели $z_1, z_2, \dots, z_8$ "соседи" центрального пикселя p . На рис. 1 представлено визуальное представление метода 8 соседей.

z_2	z_3	z_4
z_1	p	z_5
z_8	z_7	z_6

Рис.1. Изображение, представляющее метод 8 соседей

Метод карты расстояний предполагает двойное распознавание изображения. Другими словами, сначала поток $H_1(p) = \{z_1, z_2, z_3, z_4\}$ осуществляет движение справа налево снизу вверх, а затем поток $H_2(p) = \{z_5, z_6, z_7, z_8\}$ проследует слева направо сверху вниз.

Дальнейшие вычисления осуществляются следующим образом:

$$n(p, z) = \begin{cases} 2R_x(z) + 1, & \text{если } z \in \{z_1, z_5\} \\ 2R_y(z) + 1, & \text{если } z \in \{z_3, z_7\} \\ 2(R_x(z) + R_y(z) + 1), & \text{если } z \in \{z_2, z_4, z_6, z_8\}, \end{cases}$$

$$K(p, z) = \begin{cases} (1, 0), & \text{если } z \in \{z_1, z_5\} \\ (0, 1), & \text{если } z \in \{z_3, z_7\} \\ (1, 1), & \text{если } z \in \{z_2, z_4, z_6, z_8\}, \end{cases}$$

где $R(z)$ – вектор, обозначающий расстояние до ближайшего пикселя фона, причем $R(z) = (R_x, R_y)$; $n(p, z)$ – разность евклидовых расстояний от p до z , причем $z \in H_1 \cup H_2$; $K(p, z)$ – разность координат p и z , причем $z \in H_1 \cup H_2$.

Вычисления градиента карты расстояний на следующем шаге позволяет отыскивать вершины специального графа G . Если $grad \rightarrow 0$, то данная точка и есть вершина графа G [2].

Ниже представлен код для реализации данного этапа на языке C++:

```
const sd00 = map_d.getData(i - 1, j - 1);
const sd10 = map_d.getData(i, j - 1);
const sd20 = map_d.getData(i + 1, j - 1);
const sd01 = map_d.getData(i - 1, j);
const sd11 = map_d.getData(i, j);
const sd21 = map_d.getData(i + 1, j);
const sd02 = map_d.getData(i - 1, j + 1);
const sd12 = map_d.getData(i, j + 1);
const sd22 = map_d.getData(i + 1, j + 1);
const dx = (sd00 - sd11) + (sd01 - sd11) + (sd02 - sd11) +
            (sd11 - sd20) + (sd11 - sd21) + (sd11 - sd22);
const dy = (sd00 - sd11) + (sd10 - sd11) + (sd20 - sd11) +
            (sd11 - sd02) + (sd11 - sd12) + (sd11 - sd22);
gradient_x.setData(i + j * size | 0, dx);
gradient_y.setData(i + j * size | 0, dy);
```

Чтобы избавиться от избыточности вершин необходимо центрировать траекторию движения объекта относительно дороги. Для этого экспериментальным путем подбирается значение, лежащее в пределах значения вектора, а именно $[-1; 1]$.

Таким образом, двойное распознавание изображения в рамках метода карты расстояний можно представить в виде следующего алгоритма [4]:

1. Первый поток

Если $p \in M$, тогда

присваиваем $Q(p) = \infty$

Для $z = z_1, z_2, z_3, z_4$

вычисляем $Q(p) = \min(Q(p), Q(z) + n(p, z))$

Если существует такой z , для которого

$(Q(z) + n(p, z)) \rightarrow \min$ и $(Q(z) + n(p, z) < Q(p))$,

$R(p) = R(z) + K(p, z)$.

2. Второй поток

Если $p \in M$, тогда

для z_5, z_6, z_7, z_8

вычисляем $Q(p) = \min(Q(p), Q(z) + n(p, z))$

Если существует такой z , для которого

$(Q(z) + n(p, z)) \rightarrow \min$ и $(Q(z) + n(p, z) < Q(p))$,

$R(p) = R(z) + K(p, z)$

$E(p) = \sqrt{Q(p)}$.

Где Q – бинарное изображение; M – пиксели объектов.

Для построения ребер графа G необходимо перебрать все пары вершин множества вершин $V(G)$ графа G , и соединить только в том случае, если в зоне видимости располагаются обе вершины. Для формирования данной зоны выполняется трассировка, другими словами,

производится выборка l раз, где l – длина ребра, соединяющего вершины. Для избегания избыточности оставляем только катеты или наиболее короткие ребра треугольников.

Реализация данного метода

На рисунке 2 представлен снимок местности со спутника.

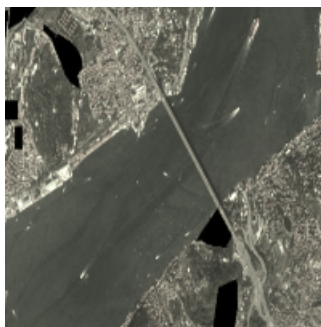


Рис. 2. Карта местности (снимок со спутника)

Согласно шагу 1 алгоритма, представленного в данной статье, необходимо преобразовать исходное изображение в бинарное. На рисунке 3 представлено получившееся изображение после преобразования.

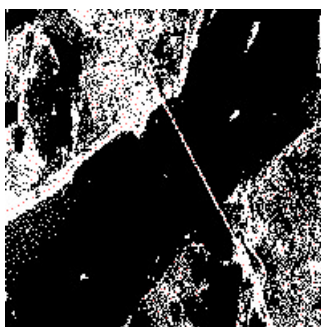


Рис. 3. Бинарное изображение местности

Для нахождения наикратчайшего пути между двумя пунктами, воспользовавшись методом карты расстояний, получим изображение, на котором представлены все возможные вершины графа G , а также ребро с минимальным расстоянием между пунктами. Данное изображение представлено на рисунке 4.

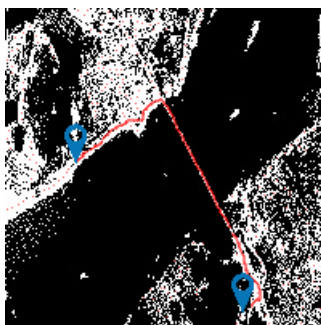


Рис. 4. Наикратчайший путь между двумя пунктами, найденный в ходе реализации метода карты расстояний

Выводы

Использование метода карты расстояний при проектировании движения беспилотных наземных устройств при помощи классического подхода на графах отличается своей точностью и наглядностью, что немало важно при решении задач такого типа. Данный алгоритм позволяет найти наикратчайший путь между двумя пунктами, учитывая расположение препятствий. Вероятность появления ошибки в карте при таком подходе минимальна, что достаточно важно в вопросах движения машин или других наземных беспилотных устройств.

Библиографический список

1. Васильев В.Е., Морозов А.В. Компьютерная графика, Санкт-Петербург, 2005. - 101 с.
2. Зыков А.А. Основы теории графов. М.:Вузовская книга, 2004. - 664 с.
3. Конакова И.П., Пирогова И.И. Инженерная и компьютерная графика: учебное пособие. Уральский университет - УрФУ, 2014. - 90 с.
4. Новиков Ф.А. Дискретная математика для программистов. Питер, 2009. - 384 с.
5. Тимофеева Г.А. Теория механизмов и механика машин. МГТУ им. Н. Э. Баумана, 2017. - 567 с.

УДК 65.011.56; ГРНТИ 50.51.02

ИСПОЛЬЗОВАНИЕ МЕТОДОЛОГИИ ARIS ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ УПРАВЛЕНИЯ БЛАГОУСТРОЙСТВА ГОРОДА РЯЗАНИ

Ю. Барышникова

*Рязанский государственный радиотехнический университет,
Российская Федерация, г. Рязань, baryshnikova1996@mail.ru*

Аннотация. В данной работе описаны основные этапы проектирования информационной системы для автоматизации деятельности Управления благоустройства города Рязани с использованием методологии ARIS.

Ключевые слова. Благоустройство города, проектирование, моделирование и анализ бизнес-процессов, информационная система

THE USE OF ARIS METHODOLOGY TO INCREASING AN EFFICIENCY OF RYAZAN URBAN AMENITIES PROVISION DEPARTMENT

J. Baryshnikova

*Ryazan state radio engineering university,
Russian Federation, Ryazan, baryshnikova1996@mail.ru*

Abstract. This paper deals with the basic steps of project designing the informational system for activity automatization of Ryazan urban amenities provision department through ARIS-methodology.

Keywords. Provision of urban amenities, project design, business process modeling and analysis, informational system

Введение

В настоящее время наряду с развитием государственного сектора в области благоустройства происходит стремительное развитие информационных технологий по совершенствованию механизмов государственного управления. Внедрение информационных технологий в области государственного и муниципального управления позволяет в значительной степени повысить эффективность выполняемых работ и мероприятий, решать большое число проблем, а также повысить качество и оперативность взаимодействия органов власти с населением.

Аналитическая деятельность в IT сфере для проектирования информационных систем приобретает в настоящее время все большее значение для решения глобальных задач повы-

шения эффективности не только отдельных предприятий, но и всего государства в целом. С течением времени все больше появляется подходов к бизнес-анализу, но одним из универсальных является использование методологии ARIS.

В настоящий момент в Управлении благоустройства города Рязани существует проблема эффективной связи с гражданами для решения возникающих вопросов или проблем, связанных с состоянием города в части благоустройства.

Целью данной работы является проектирование информационной системы для повышения эффективности работы Управления и повышения качества и оперативности взаимодействия данной организации с населением.

Принципы методологии ARIS

Методология ARIS предназначена для бизнес-анализа и проектирования информационных систем. Акроним ARIS (с англ. Architecture of Integrated Information System) подразумевает под собой также интегрированное средство моделирования, разработанное германской фирмой IDS Scheer.

ARIS - современная методология бизнес-моделирования, которая позволяет описывать, анализировать и оптимизировать различные аспекты деятельности организаций посредством совокупности групп моделей, отвечающие вопрос:

- 1) Оргструктура – инфраструктура и организационная структура («Кто?»);
 - 2) Функции – функциональная деятельность («Что?»);
 - 3) Информация – используемые организацией данные в процессе ее функционирования («На основе чего?»);
 - 4) Процессы – бизнес-процессы организации («Каким образом?»);
 - 5) Входы/выходы – результаты, на которые направлена деятельность («Для чего?»)
- [1].

В результате анализа и проектирования по данной методологии визуализируется необходимая для информационной системы архитектура организации, позволяющая увидеть и оптимизировать деятельность организации в целом.

В связи с большим числом вариаций применяемых по каждому из аспектов моделей для каждой организации может быть применен свой подход в выборе необходимых моделей. Наиболее удобные для проведения анализа модели приведенные в таблице ниже (Таблица 1).

Таблица 1. Модели нотации ARIS

Модель	Аспект анализа
BPMN 2.0	Процессы
Organizational char	Оргструктура
ER-model	Информация
Function tree	Функции

Повышение эффективности деятельности Управления благоустройства города Рязани

Выделим четыре наиболее приоритетные направления деятельности Управления, связанные со взаимодействием с населением, для проектирования информационной системы по методологии ARIS:

- 1) проведение мероприятий по общегородскому благоустройству территорий;
- 2) своевременное рассмотрение обращений граждан по вопросам, относящимся к сфере деятельности Управления;
- 3) прием проектов по благоустройству городской территории от жителей города;
- 4) обеспечение населения информацией о деятельности по благоустройству на территории города Рязани.

Повышение эффективности именно в этих четырёх областях деятельности зачёт информационных технологий – оптимальное решение в рамках эпохи цифровой экономики, когда связь с населением устанавливается через оперативное предоставление необходимых услуг для граждан.

Разработка архитектуры ARIS

Первый этап разработки архитектуры ARIS – построение, сбор метрик и анализ моделей бизнес процессов «Как есть» деятельности Управления. Для анализа в данной работе использовалась метрика «Уровень автоматизации» в качестве оценки эффективности. Метрика вычисляется соотношением количества автоматизированных действий к сумме всех действий в бизнес процессе.

На основе полученных данных строятся модели бизнес-процессов «Как есть», нацеленные на улучшение метрик, полученных на первом этапе [2].

Анализ метрик бизнес-процессов «Как должно быть» выбранных бизнес-процессов Управления благоустройства города Рязани показал, что текущая деятельность автоматизирована на 0% по каждому из четырёх направлений. В бизнес процессах «Как должно быть» уровень автоматизации значительно увеличился (рис. 1Рис.).

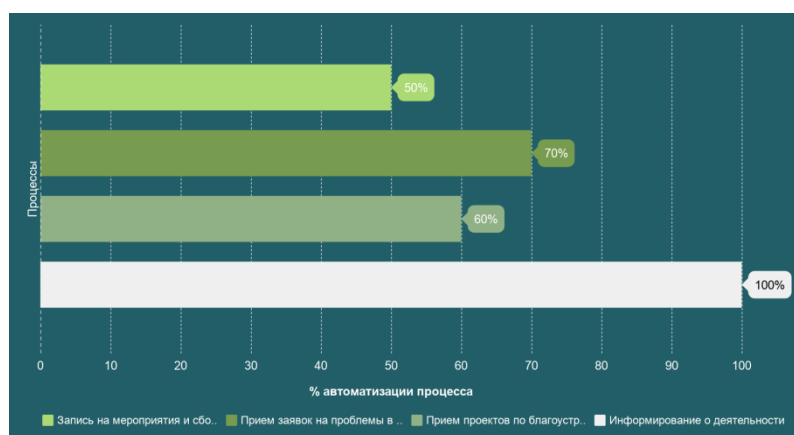


Рис. 1. Увеличение уровня автоматизации в процессах «Как должно быть», %

На основе анализа моделей «Как должно быть» детализировано строятся модели организационной структуры, данных и функций [1].

На последнем этапе происходит объединение полученных моделей в единое целое. Полученная архитектура ARIS позволяет оценить кто (организационная модель) что (модель функций) делает в организации, для чего (модель выходов – собираются все выходные документы из бизнес-процессов), на основе чего (модель данных) и каким образом (модель бизнес-процессов) осуществляется деятельность (рис. 2).

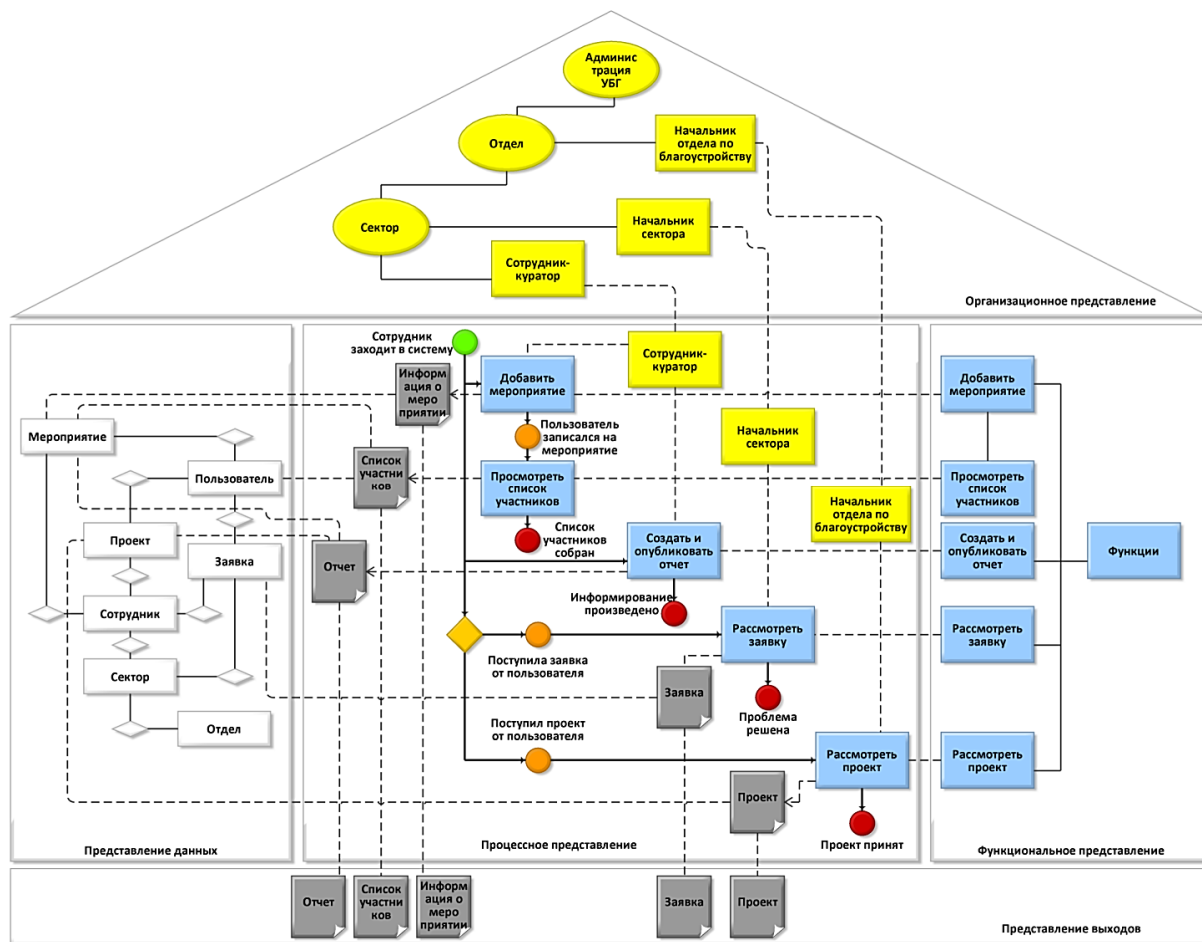


Рис. 2. Архитектура ARIS

Выводы

Наиболее оптимальным решением задачи повышения эффективности деятельности Управления благоустройства города Рязани является модификация бизнес-процессов и достижение автоматизации деятельности путем внедрения информационной системы. Существуют методология, которая позволила спроектировать работу организации с подобной системой, а именно, методология ARIS, позволяющая рассмотреть организацию сразу с пяти сторон: деятельность, организационная структура, данные, функции и продукты.

Так, система, спроектированная по данной методологии, и которую можно внедрить в Управление, позволяет автоматизировать деятельность по четырем направлениям: прием заявок на проблемы, прием проектов по благоустройству, информирование населения о результатах деятельности и проведение мероприятий.

Библиографический список

1. Инструментарий ARIS. Методы – Весть – МетаТехнология, Апрель 2000 – 227.
2. Шеер А.В. Бизнес-процессы. Основные понятия. Теория. Методы – Весть – МетаТехнология, 2-е издание, 1999-173.

УДК 347.736; ГРНТИ 10.23.47

ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННО-КОММУНИКАТИВНЫХ ТЕХНОЛОГИЙ В ОБЛАСТИ ПРОГНОЗИРОВАНИЯ БАНКРОТСТВА ПРЕДПРИЯТИЙ

Ю. Маркина

*Рязанский государственный радиотехнический университет,
Россия, Рязань, ulechkamarkina@mail.ru*

Аннотация. Данная работа посвящена анализу финансового состояния предприятия на примере ООО «Ашан». Были разработаны методика и алгоритм, с помощью которых можно спрогнозировать вероятность банкротства организаций, в том числе и попавших в зону неопределенности. Предложенный в работе метод с использованием информационных технологий позволит владельцам и руководителям различных компаний осуществить прогноз на дальнейшее развитие организации и выявить динамику ее финансового состояния.

Ключевые слова. Банкротство, прогнозирование банкротства, информационные технологии, финансовое состояние.

THE USE OF INFORMATION AND COMMUNICATION TECHNOLOGIES IN THE REGION PREDICTING BANKRUPTCY OF ENTERPRISES

J. Markina

*Ryazan State Radio Engineering University,
Russia, Ryazan, ulechkamarkina@mail.ru*

Abstract. This work is devoted to the analysis of the financial condition of the enterprise on the example of "Auchan". Were developed a technique and algorithm by which you can predict the probability of bankruptcy of organizations, including those in the zone of uncertainty. The proposed method using information technology will allow owners and managers of various companies to make a forecast for the further development of the organization and to identify the dynamics of its financial condition.

Keywords. Bankruptcy, bankruptcy forecasting, information technologies, financial condition.

Введение

За последние десять лет проблема прогнозирования банкротства предприятий является особенно актуальной в Российской Федерации. На данный факт сильное влияние оказало реформирование российской экономики и постепенное вхождение России в зону кризиса, вследствие этого на грани банкротства оказались многие организации и предприятия.[0]

На текущий момент очень важно оценить состояние предприятия, чтобы была возможность определить какова вероятность банкротства и вовремя принять меры для восстановления платежеспособности.

Основными задачами данной работы являются:

- изучение теории банкротства;
- рассмотрение различных методик для прогнозирования банкротства;
- изучение показателей финансового состояния предприятия.

Цель работы: исследование динамики вероятности банкротства предприятия для прогнозирования его дальнейшего развития с использованием информационных технологий.

Объектом анализа финансового состояния является: Общество с ограниченной ответственностью «Ашан».

Методики прогнозирования банкротства предприятия

Рассмотрим несколько из существующих методов прогнозирования банкротства предприятия:[0]

- двухфакторная модель Альтмана;
- пятифакторная модель Альтмана;
- модифицированная пятифакторная модель Альтмана.

Формула двухфакторной модели Альтмана :

$$Z = -0,3877 - 1,0736K_{\text{тл}} + 0,0579K_{\text{зс}}, \quad (1)$$

где $K_{\text{тл}}$ — коэффициент текущей ликвидности;
 $K_{\text{зс}}$ — коэффициент капитализации, высчитываются по формулам:

$$K_{\text{тл}} = \frac{\text{оборотные активы}}{\text{текущие обязательства}},$$

$$K_{\text{зс}} = \frac{\text{долгосрочные обязательства}}{\text{собственный капитал}}.$$

Интерпретация результатов:

- $Z < 0$ — вероятность банкротства меньше 50% и далее снижается по мере уменьшения Z ;
- $Z = 0$ — вероятность банкротства равна 50%;
- $Z > 0$ — вероятность банкротства больше 50% и возрастает по мере увеличения рейтингового числа Z . [0]

Позднее Альтман создал более развернутую *пятифакторную модель*. В общем виде она имеет вид:

$$Z = 1.2X1 + 1.4X2 + 3.3X3 + 0.6X4 + X5, \quad (2)$$

где $X1 = \frac{\text{оборотный капитал}}{\text{сумма активов}};$

$$X2 = \frac{\text{чистая прибыль}}{\text{сумма активов}};$$

$$X3 = \frac{\text{прибыль до налогообложения}}{\text{сумма активов}};$$

$$X4 = \frac{\text{рыночная стоимость акций}}{\text{заемный капитал}};$$

$$X5 = \frac{\text{выручка}}{\text{сумма активов}}.$$

Расчеты по модели Альтмана показали, что обобщающий показатель Z может лежать в пределах $[-14, +22]$.

В результате Z -подсчета показателя делается заключение:

- $Z < 1,81$ - предприятие признается банкротом;
- $Z \in [1,81; 2,99]$ - ситуация неопределенна;
- $Z > 2,99$ - присуще стабильным и финансово устойчивым компаниям.

Z -коэффициент имеет серьезный недостаток - использование данного коэффициента лишь в отношении крупных компаний, у которых котируются акции на биржах. Поэтому позже Альтман разработал модифицированный вариант формулы для компаний, акции которых не котируются на бирже, так называемая модифицированная пятифакторная модель Альтмана:

$$Z = 0.717X1 + 0.847X2 + 3.1X3 + 0.42X4 + 0.995X5, \quad (3)$$

где $X1 = \frac{\text{оборотный капитал}}{\text{сумма активов}};$

$$X2 = \frac{\text{чистая прибыль}}{\text{сумма активов}} ;$$

$$X3 = \frac{\text{прибыль до налогообложения}}{\text{сумма активов}} ;$$

$$X4 = \frac{\text{балансовая стоимость акций}}{\text{заемный капитал}} ;$$

$$X5 = \frac{\text{выручка}}{\text{сумма активов}} .$$

В результате Z-подсчета показателя делается заключение:

- $Z < 1,23$ - предприятие признается банкротом;
- $Z \in [1,23; 2,89]$ - ситуация неопределенна;
- $Z > 2,9$ - присуще стабильным и финансово устойчивым компаниям.

Проведем прогноз банкротства компании ООО "Ашан" помощью модифицированной пятифакторной модели, так как она предсказывает вероятность банкротства в 95% случаях.

Прогноз по модифицированной пятифакторной модели Альтмана по формуле (3):

$$Z = 0.717X1 + 0.847X2 + 3.1X3 + 0.42X4 + 0.995X5,$$

Расчеты на 2015 год показали следующий результат:

$$3,169610474 > 2,9.$$

Из данных вычислений можно сделать вывод, что компания финансово устойчива, и вероятность банкротства крайне низка.

Проведем расчет на период 31.12.14 г.:

$$1,23 < 2,788999518 < 2,9.$$

Значение числа Z попало в диапазон $[1,23; 2,9]$ (рис. 1), который соответствует зоне неопределенности, поэтому можно сделать вывод, что на период 2014 года невозможно осуществить прогноз на дальнейшее развитие компании.

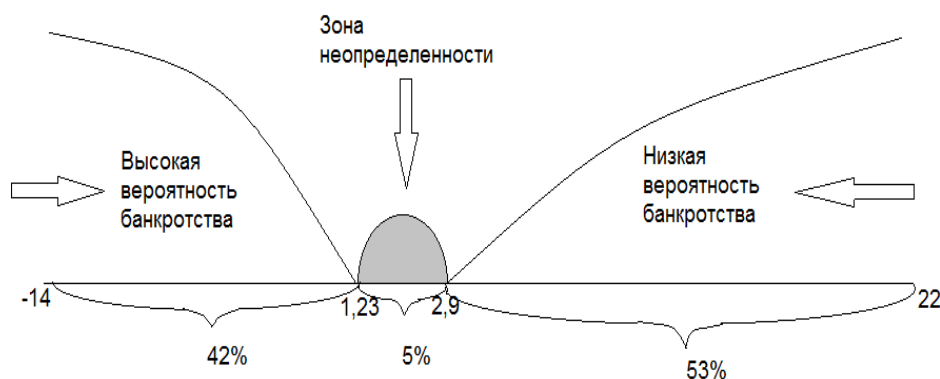


Рис. 1. Зоны вероятности банкротства

Прогнозирование банкротства с помощью информационных технологий

Кроме вышеперечисленных методик прогнозирования банкротства предприятия, для полного анализа проводится статистический анализ выборки. Это можно сделать с помощью программы SPSS Statistics, главное меню которой показано на рисунке 2.

SPSS Statistics (аббревиатура англ. «*Statistical Package for the Social Sciences*») — «статистический пакет для социальных наук») — компьютерная программа для статистической обработки данных, один из лидеров рынка в области коммерческих статистических продуктов, предназначенных для проведения прикладных исследований в социальных науках.

Основные возможности программы:

- ввод и хранение данных;
- возможность использования переменных разных типов;
- частотность признаков, таблицы, графики, таблицы сопряжённости, диаграммы;
- первичная описательная статистика;
- маркетинговые исследования;
- анализ данных маркетинговых исследований

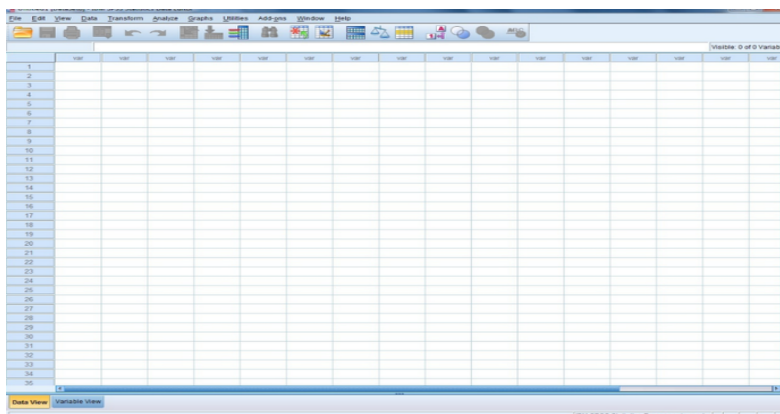


Рис. 2. Главное меню SPSS Statistic

Выводы

В данной работе были рассмотрены самые распространенные методики прогнозирования банкротства предприятия.

С помощью методики Альтмана на основе модифицированной пятифакторной модели был проведен прогноз банкротства ООО "Ашан", где были получены результаты - в период на 31.12.15 г. организация являлась финансово-устойчивой, а в период на 31.12.14 г. ситуация неопределенна.

С помощью компьютерной программы для статистической обработки данных SPSS Statistics был проведен статистический анализ, который состоял в нахождении основных числовых характеристик финансового состояния предприятия.

Таким образом, можно сделать общий вывод, что цель и задачи, поставленные в начале данной работы, были выполнены.

Библиографический список

1. Банкротство предприятия и его прогнозирование лекций [Электронный ресурс]. URL: <http://works.doklad.ru/view/CrXIM4WkYZk.html> (дата обращения 15.02.2018).
 2. "Количественные" кризис-прогнозные методики: Коэффициент Альтмана (индекс кредитоспособности) [Электронный ресурс]. URL: <http://3.scicenter.online/management/532kolichestvennyie-krizisprognoznyie.html> (дата обращения 15.02.2018).
 3. Банкротство предприятия. Понятия и причины банкротства [Электронный ресурс]. URL: <http://www.grandars.ru/college/pravovedenie/bankrot.html> (дата обращения: 15.02.2018).
- УДК 681.391; ГРНТИ 47.05.17

ОБЗОР СОВРЕМЕННЫХ МЕТОДОВ ПОМЕХОУСТОЙЧИВОГО КОДИРОВАНИЯ

Е.П. Королева

*Рязанский государственный радиотехнический университет,
Россия, Рязань, ap.koroleva@bk.ru*

Аннотация. В данной работе рассмотрены современные методы помехоустойчивого кодирования, а именно сверточное кодирование, турбо кодирование, кодирование Рида-Соломона, кодирование с малой плотностью проверок на четность. Проанализированы преимущества и недостатки использования этих методов.

Ключевые слова. Помехоустойчивое кодирование, сверточные коды, блочные коды

REVIEW OF MODERN METHODS ERROR-CORRECTING CODING

E.P. Koroleva

*Ryazan State Radio Engineering University,
Russia, Ryazan, ap.koroleva@bk.ru*

Abstract. This paper deals with the modern methods of the error-correcting coding: convolutional coding, turbo coding, Reed-Solomon coding, coding with low density parity checks. The advantages and disadvantages of using these methods are analyzed.

Keywords. Error-correcting coding, convolution codes, block codes

В настоящее время системы передачи данных распространены практически во всех сферах деятельности человека. Однако при передаче данных на дальние расстояния сообщения часто принимаются в искаженном виде. Для повышения достоверности передаваемых сообщений используют помехоустойчивое кодирование. Существует огромное множество кодов, которые применяются в различных каналах и обладают разными свойствами.

Помехоустойчивые коды делятся на две большие группы.

Блочные коды, где кодируемая последовательность разбивается на отдельные блоки с добавлением к каждому блоку определенного числа проверочных символов. Кодирование и декодирование осуществляются в пределах этого блока, который называют кодовой комбинацией).

Непрерывные/цепные/рекуррентные/сверточные коды. Операция кодирования и декодирования осуществляется над непрерывной последовательностью символов, то есть без деления на блоки, как происходит в блочных кодах, при этом проверочные символы размещают в определенном порядке между символами сообщения. Такие коды позволяют допускать к меньшему расширению полосы пропускания.

Наиболее известными в блочных кодах являются циклические коды – это коды, которые выполняют условия цикличности: все строки матрицы, которой можно представить любой групповой код, могут быть получены циклично путем сдвига образующей комбинации кода. [1]

Рассмотрим наиболее популярные 4 типа помехоустойчивого кодирования в системах радиосвязи.

Сверточное кодирование

Такие коды позволяют допустить меньшее расширение полосы пропускания относительно других кодов. Характеристики описываются тремя параметрами: n/k – скорость кодирования, то есть информация, которая приходится на один закодированный бит, l – длина кодового ограничения (на практике является небольшим целым числом), которая отражает число разрядов k -кортежа в кодирующем регистре сдвига (изменяется для контроля мощности и сложности кода).

Под корректирующей способностью кода понимают гарантируемое число ошибок в кодовых комбинациях, которые будут исправлены заданным кодом.

Правило кратности исправляемых ошибок в сверточных кодах: ошибки исправляются в случае, если в последовательности произошло не более g_i ошибок, причем g_i определяется выражением

$$2 * g_i + 1 \leq D_i, \text{ где } D_i - \text{минимальное кодовое расстояние.}$$

С минимальным кодовым расстоянием связана вероятность исправления ошибок при декодировании.

Под энергетическим выигрышем кода (далее – ЭВК) η понимается выигрыш по помехоустойчивости при применении корректирующего кодирования. Чтобы получить заданную по условию вероятность ошибочного приема символа в информационной последовательности необходимо обеспечить минимально допустимое отношение сигнал/шум на входе демодулятора.

При передаче информации за заданное время требуется передать k информативных символов в случае если мы не используем корректирующее кодирование. В противном случае за это же время необходимо также передать n символов (информативные и проверочные) при том же уровне сигналов. Для этого сокращают длительность символов при передаче, однако это требует расширения полосы частот в n/k раз. А из этого следует, что значение вероятности будет обеспечиваться при отличном от первоначального отношения сигнал/шум. Именно эта разница отношений сигнал/шум (при применении кодирования) и сигнал/шум (без использования кодирования) при положительном значении указывает на ЭВК (выражается в децибелах). ЭВК может быть использован, например, путем увеличения скорости передачи данных, уменьшения антенн и мощности передатчиков. Наиболее выгодными для ЭВК являются сверточные коды с малой длиной кодового ограничения и с декодированием по алгоритму Витерби.

Сложность кодера (на стороне передачи) линейно зависит от длины регистра и кодового ограничения. Декодеры просты при исправлении ошибок невысокой кратности. Сложность декодера (на стороне приема) растет пропорционально максимальному числу выходных символов, на которые влияет один информационный символ (полной длине кодового ограничения), а также определяется методом декодирования сверточных кодов:

- пороговое декодирование (мажоритарный метод) – наиболее простой в реализации как блочных, так и сверточных кодов;
- последовательное декодирование;
- декодированием по алгоритму Витерби – сложен в реализации, так как объем вычислений растет экспоненциально относительно увеличения длины кодового ограничения.

Важными характеристиками кодирования являются: надежность (вероятность корректного декодирования информационной последовательности) и трудоемкость (число вычислительных операций, которые были затрачены на декодирование одного блока символов из принимаемой последовательности). [2]

В зависимости от параметров и характеристик сверточных кодов различают несколько их разновидностей:

а) *систематические*, когда в последовательности кодовых символов можно отделить избыточные символы от информационных. Эти коды позволяют получить на приемной стороне оценку информационных символов, не производя декодирования или иной обработки принимаемых символов.

б) *несистематические* коды, преимущество которых является относительно большее минимальное свободное расстояние, по сравнению с систематическими кодами при прочих равных условиях.

в) *катастрофические сверточные* коды при которых конечное число ошибок, приобретенных в канале связи, может привести к бесконечному числу ошибок в декодированных символах. Катастрофический сверточный код будет порождаться кодером, когда на его диаграмме состояний будет замкнутый путь нулевого веса, соответствующий ненулевым кодируемым информационным символам.

г) *прозрачные сверточные* коды исправляют независимые ошибки в канале с фазовой неоднозначностью. Кодер должен отвечать условиям инвариантности (или прозрачности) по отношению к операции инвертирования кода.

д) *перфорированные сверточные* коды, кодовые последовательности которых могут быть получены путем периодического вычеркивания (перфорации) символов из комбинации.

Кодирование Рида-Соломона

Код допускает небольшое расширение полосы пропускания, может указывать на наличие неисправимых ошибок, используется при работе с цифровой коммуникацией и хранением информации. Также позволяет исправлять ошибки в блоках данных, поэтому он используется в:

- запоминающих устройствах;
- спутниковой связи;
- цифровом телевидении;
- беспроводной и мобильной связи;
- вывocosкоростных модемах (ADSL, xDSL, и т.д.)

Кодирование и декодирование осуществляется программным путем в ПЛИС или специальных кодерах. Кодер добавляет избыточные биты в цифровой блок данных. При передаче и хранении информации возникают ошибки: помехи, наложение, физические повреждения съемных носителей данных и т.д. Декодер пытается вернуть данные в оригинальном виде, исправляя ошибки при обработке каждого блока данных. В зависимости от характеристик этого кода варьируется количество и тип ошибок, которые можно исправить.

Код обозначается $R_s(n, k)$ с символами, состоящими из s бит. Кодер берет k символов данных (по s бит в каждом блоке) и добавляет проверочные символы до длины n , получая кодовое слово. Декодер при ошибке в кодовом слове может исправить до $t = (n - k) / 2$ символов и путем алгебраических процедур декодирования до $2t$ стирающих сигналов (стирающий сигнал генерируется, когда позиция ошибки символа известна).

Демодулятор «помечает флагами» символы, которые, возможно, содержат ошибки. После декодирования принятого слова есть 3 возможных исхода:

- а) $2s + r < 2t$ (s – ошибки, r – стирающие сигналы). На выходе декодера переданное слово будет восстанавливаться;
- б) сигнал от декодера о невозможности восстановить переданное кодовое слово;
- в) неисправимая ошибка – возвращается некорректное кодовое слово без указания этого факта (декодер не исправил возникшие ошибки).

Достоинство кода Рида-Соломона – как правило вероятность ошибки в декодированных данных намного ниже, чем в системе без кодирования. Сообщение можно представить в виде набора символов некоторого алфавита. Кодовое слово формируется с помощью специального многочлена. [3]

Турбо-кодирование

Код допускает большое расширение полосы пропускания, однако такое кодирование считается очень эффективным при отсутствии требований к полосе.

Турбо-кодом можно назвать параллельный каскадный блочный систематический код, который состоит из каскада параллельно соединенных систематических кодов, которые называются компонентными кодами. К числу наилучших из компонентных кодов относят коды Хемминга, код Боуза-Чоудхури-Хоквингема (далее – БЧХ), Рида-Соломона и другие.

Турбо-коды являются высокоэффективным помехоустойчивым кодом с коррекцией ошибок. Их применяют в цифровой связи, электротехнике, спутниковой связи, а также в областях, связанных с поиском способов достижения максимальной скорости передачи данных по каналу связи с шумами в ограниченной полосе частот.

Преимущества турбо-кодов:

- коды наиболее близки к границе Шеннона, которая является теоретическим пределом максимальной пропускной способности зашумленного канала;
- коды увеличивают скорость передачи информации хотя и не требуют увеличения мощности передатчика, поэтому они используются для уменьшения мощности при передаче с заданной скоростью;
- сложность декодирования не зависит от длины информационного блока. Это позволяет в помощь увеличения длины сообщения снизить вероятность ошибки декодирования.

Недостатки турбо-кодов:

- высокая сложность декодирования;
- большая задержка;
- небольшое кодовое расстояние. При большой входной вероятности ошибки эффективной кода высока, а при малой – крайне ограничена.

Из-за выявленных недостатков для уменьшения вероятности ошибок используют коды с малой плотностью проверок на четность (LDPC-коды).

В основе турбо-кодирования лежит использование двух параллельно работающих элементарных кодеров. Информационный блок кодируется дважды, а при декодировании блок можно разделить на два кодовых блока. Это позволяет использовать два декодера, которые обрабатывают каждый свой кодовый блок.

Принцип итеративного (или турбо) декодирования состоит в том, что декодированную информацию с выхода одного декодера используют как входную информацию на другом, причем эту операцию можно повторять многократно.

В турбо-кодах для повышения эффективности используют перемешивание поступающих бит по псевдослучайному закону. Задача заключается в том, чтобы преобразовать входную последовательность так, чтобы комбинации бит, имеющие наименьший вес (вес – число ненулевых бит кодового слова), на выходе первого декодера были преобразованы в комбинации с наибольшим весом на других кодерах. При кодировании кодовые слова формируются так, чтобы среднее расстояние между ними было максимальным (число бит, в которых различаются кодовые комбинации). На выходе турбо-кодера среднее расстояние между кодовыми словами больше, чем минимальное расстояние для каждого отдельного кодера, следовательно растёт эффективность кодирования. [4]

Кодирование с малой плотностью проверок на четность (Low-density-paritycheck code, далее код LDPC)

Код допускает небольшое расширение полосы пропускания, может указывать на наличие неисправимых ошибок. Коды обладают наилучшим минимальным расстоянием, однако при довольно простом декодировании обеспечивают высокую степень исправления ошибок. При увеличении длины LDPC-коды превосходят турбо-коды и приближаются по пропускной способности к аддитивному белому гауссовскому шуму (далее – АБГШ). Многие конструкции LDPC-кодов являются циклическими или квазициклическими, это позволяет ис-

пользовать быстрое декодирование и эффективные процедуры кодирования. Основными конкурентами являются турбо-коды, однако LDPC-коды имеют некоторые преимущества:

- LDPC-коды быстрее по скорости декодирования;
- LDPC-коды предпочтительнее в каналах с меньшими вероятностями ошибок.

В настоящее время улучшаются каналы передачи в связи с развитием методов передачи информации, что способствует развитию LDPC-кодов. Применяя методы итеративного декодирования к кодам, можно наиболее близко приблизиться к пропускной способности канала при несложной реализации. LDPC-коды рекомендованы для исправления ошибок к стандартам передачи данных DVB-S2, 802.11n, 802.16e.

LDPC-коды востребованы в системах передачи данных, при условии максимальной скорости передачи и ограниченной полосе частот. К таким системам относят цифровое телевидение высокой четкости, спутниковую связь. LDPC-кодеры в этих системах обеспечивают огромную скорость передачи данных (до 40 Гб/с) в связи с простой реализацией.

Наиболее быстрыми декодерами принято считать многопороговые декодеры (далее – МПД). В них можно декодировать длинные коды, выполняя простейшие операции сложения и сравнения небольших чисел. МДП крайне просты в аппаратной и программной реализации, например, могут быть реализованы с использованием линейных сдвиговых регистров, которые являются самыми быстрыми аппаратными элементами. [5]

Довольно часто на практике применяется последовательное кодирование, то есть комбинированное, где в качестве внутреннего кода в системе может использоваться один LDPC-код, а в качестве внешнего один сверточный.

Выводы

Среди рассмотренных методов помехоустойчивого кодирования нельзя выделить один самый эффективный, так как они применимы в совершенно разных условиях и при различных начальных данных системы. Помехоустойчивое кодирование не стоит на месте и с каждым днем развивается, открывая новые перспективы передачи данных без помех, шумов и ошибок.

Библиографический список

1. Помехоустойчивость и эффективность систем передачи информации / А. Г. Зюко, А. И. Фалько, И. П. Панфилов, Л. В. Банкет, П. В. Ивашенко; Под ред А. Г. Зюко. - М: Радио и связь, 2005 - 272 с..
2. TM Synchronization and Channel Coding, Recommendation for Space Data System Standards CCSDS 131.0-B-1, Issue 1, Blue Book, Consultative Committee for Space Data Systems, September, 2003.
3. Никитин Г.И. Сверточные коды. Учеб. пособие. — СПб.: Издательство СПбГУАП, 2001.
4. Золотарёв В.В., Овечкин Г.В. Эффективные алгоритмы помехоустойчивого кодирования для цифровых систем связи // Электросвязь. 2003, №9. С.34-37.
5. Золотарёв В.В., Овечкин Г.В. Помехоустойчивое кодирование. Методы и алгоритмы. Справочник. М.: «Горячая линия – Теле-ком», 2004. 124 с

СЕКЦИЯ «СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ»

УДК 621.316 : 681.518; ГРНТИ 44.29.31, 28.17.31

ФУНКЦИОНАЛЬНО-БЛОЧНОЕ МОДЕЛИРОВАНИЕ СИСТЕМЫ ЗАЩИТЫ ОТ ПЕРЕГРУЗОК ЦИФРОВЫХ ПОДСТАНЦИЙ МЭК 61850

А.С. Войнов*, В.Н. Дубинин*, И.В. Сенокосов, Л.П. Климкина**, В.В. Вяткин***

* Пензенский государственный университет,
Россия, Пенза, duvini@mail.ru

** Пензенский государственный аграрный университет,
Россия, Пенза, ludmila.klimkina@gmail.com

*** Технический университет Лүлео,
Швеция, Лүлео, valeriy.vyatkin@ltu.se

Аннотация. Представлена визуальная имитационная модель системы защиты от перегрузок по току цифровых подстанций МЭК 61850. Данная модель реализована на основе функциональных блоков стандарта МЭК 61499 в системе NxtStudio, что позволяет использовать ее без значительных переделок в реальных системах защиты. В работе также предлагается структура программного обеспечения для генерации функционально-блочных моделей защиты от перегрузок.

Ключевые слова. Сети электроснабжения, цифровая подстанция, защита от перегрузок, стандарт МЭК 61850, стандарт МЭК 61499, имитационная модель, функциональный блок, NxtStudio.

FUNCTION BLOCK-BASED MODELLING OF IEC 61850 SUBSTATIONS OVERCURRENT PROTECTION

A.S.Voinov*, V.N. Dubinin*, I.V. Senokosov*, L.P. Klimkina**, V.V.Vyatkin***

* Penza State University,
Penza, Russia, duvini@mail.ru

** Penza State Agricultural University,
Penza, Russia, ludmila.klimkina@gmail.com

*** Lulea University of Technology,
Sweden, Lulea, valeriy.vyatkin@ltu.se

Abstract. A visual simulation model of IEC 61850 substations overcurrent protection is presented. This model is implemented on the basis of IEC 61499 function blocks in NxtStudio that allows using it without significant alterations in real protection systems. The paper also proposes a software structure for generating a function block-based overcurrent protection model.

Keywords. Electrical grid, substation, overcurrent protection, IEC 61850, IEC 61499, simulation model, function block, NxtStudio.

В настоящее время в области промышленного электроснабжения происходят значительные изменения. Во-первых, это связано с постоянно возрастающим числом как потребителей, так и производителей электроэнергии. Наблюдается значительный рост источников альтернативной энергии – ветряных, солнечных и приливных электростанций, а также узлов накопления электроэнергии, содержащих аккумуляторные батареи большой емкости. Возникает необходимость интеграции разнородных источников электроэнергии в единую сеть. Каждому пользователю должно быть предоставлено право участвовать в генерации и распространении электроэнергии. Во-вторых, значительно возрастают требования как к стоимости, так и качеству поставки электроэнергии, поскольку любое отклонение от нормы может привести к серьезным сбоям на предприятии, что неизбежно приведёт к значительным убыткам [1]. Данные требования привели к появлению умных сетей электроснабжения (*Smart Grid*) [2]. Большую роль для построения сетей *Smart Grid* играет международный стандарт МЭК 61850 [3]. Это универсальный стандарт, который позволяет упорядочить разрозненные

решения различных производителей устройств релейной защиты и систем передачи данных, применяемых на подстанциях.

Схемы защиты позволяют избежать остановки производственных или других процессов, вызванной критическим изменением напряжения (слишком высокое или низкое) или ошибками короткого замыкания. В работе [4] рассматриваются различные пути решения проблемы защиты от перегрузок по току. Выделяется два типа ошибок: ошибки на генераторе электроэнергии и ошибки на общей шине. Рассматриваются следующие схемы защиты: 1) защита распределительной шины; 2) выборочное отключение резервирования; 3) схема симпатического (индуцированного) отключения; 4) защита от отказа выключателя. Симпатическое отключение является феноменом в системах электроснабжения, когда устройство защиты в «здоровой» секции системы обрабатывает отказ в другой секции системы. Симпатическое отключение приводит к ненужному отключению электропитания работоспособного оборудования. В работе [4] приводится общее описание модели для тестирования, но не приводится конкретный пример. Это определяет актуальность разработки имитационных моделей для анализа и тестирования схем защиты.

В качестве основы для построения такой имитационной модели была принята работа [5]. В данной работе рассматривается система защиты от перегрузок на подстанции, реализующая сценарий распределенной защиты от симпатического отключения на основе стандартов МЭК 61850 и МЭК 61499. Основными артефактами проектирования в стандарте МЭК 61499 являются функциональные блоки (ФБ) [6]. В работе [5] реализация защиты на цифровой подстанции (ЦПС) МЭК 61850 выполнена на основе ФБ, однако представлена только сеть ФБ верхнего уровня. Кроме того, не разработан визуальный человеко-машинный интерфейс (ЧМИ) для тестирования схемы защиты. Целью нашей работы является развитие работы [5] в направлении реализации визуальных имитационных моделей систем защиты от перегрузок по току цифровых подстанций МЭК 61850, а также разработка программных средств для автоматической генерации этих моделей.

Логическая структура схемы защиты от перегрузок (назовем ее элементом защиты) представляет собой комбинацию из четырех логических узлов: *TCTR* – датчик, который считывает текущее значение величины силы тока в измеряемом устройстве; *PIOC* – сравнение значения, полученного от датчика, с пороговым; *PTRC* – логический узел, который отвечает за рассылку сообщения об ошибке другим устройствам защиты в сети; *XCBR* – выключатель для разрыва соединения. Полная структура данной схемы защиты представлена в [4].

Кратко рассмотрим алгоритм ее работы. Датчик (*TCTR*) постоянно считывает текущее значение величины силы тока в заданной линии и выдает соответствующий сигнал *TCTR.AmpSV_instMag_f*. При поступлении этого значения в *PIOC* происходит его сравнение с некоторым внутренним значением, представляющим собой порог. Если текущее значение тока превышает пороговое, то выдаётся сигнал *PIOC.Op_general*. Следует заметить, что в *PIOC* также поступают внешние сигналы *PTRC.Tr_general*, которые свидетельствуют о том, что другими элементами защиты обнаружена ошибка. В этом случае также выдаётся сигнал *PIOC.Op_general*. Логический узел *PTRC* отвечает за проверку внутренних параметров при поступлении сигнала *PIOC.Op_general*. В нашей работе принято, что внутренний параметр только один – время действия сигнала *PIOC.Op_general*. Если *PIOC.Op_general* слишком долго находится в активном состоянии, то происходит выдача сигнала *PTRC.Tr_general* (как на узел *XCBR*, так и наружу - другим внешним элементам защиты). Последним узлом структуры является узел *XCBR*. Данный логический узел производит разрыв цепи при поступлении на него сигнала *PTRC.Tr_general* (в том числе и извне).

Простая распределительная сеть с реализованной распределённой схемой защиты от симпатического отключения представлена на рисунке 1. В ней имеется один источник электроэнергии (сверху) и три потребителя (внизу). Всего в данной сети используется четыре ин-

теллектуальных электронных устройства защиты (*PIED* - *Protective Intelligent Electronic Devices*), которые соединены между собой и могут обмениваться сообщениями.

Система защиты от перегрузок, включающая логические узлы *TCTR*, *PIOC*, *PTRC*, *XCBR*, была реализована в виде системы ФБ (рис. 2), причем использовалась инструментальная система *NxtStudio* [7]. При тестировании системы защиты на вход *VALUE* блока *TCTR* подается случайное число в диапазоне от 10 до 100, определяющее условную силу тока. При моделировании приняты следующие допущения: 1) превышением порога считается сила тока более 90; 2) для обнаружения ошибки данное превышение должно сохраняться в течение 1,5 сек. (при условии измерения значения датчиком каждые 1 сек.). Данные условия позволяют пошагово, в визуальном режиме промоделировать работу схемы.

В ходе работы было выявлено два возможных случая срабатывания системы защиты:

1) если на одном элементе превышение порогового напряжения сохраняется более 1,5 сек.; 2) если последовательно на нескольких элементах выявлено превышение порогового напряжения и суммарная длительность превышения в системе более 1,5 сек.

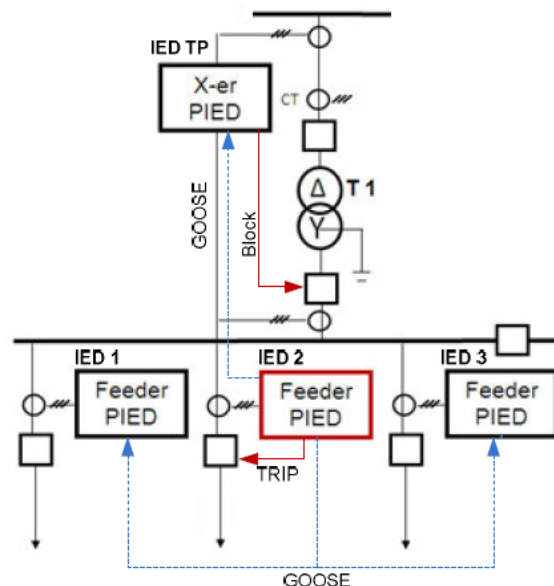


Рис. 1. Простая распределительная сеть с реализованной распределённой схемой защиты от симпатического отключения

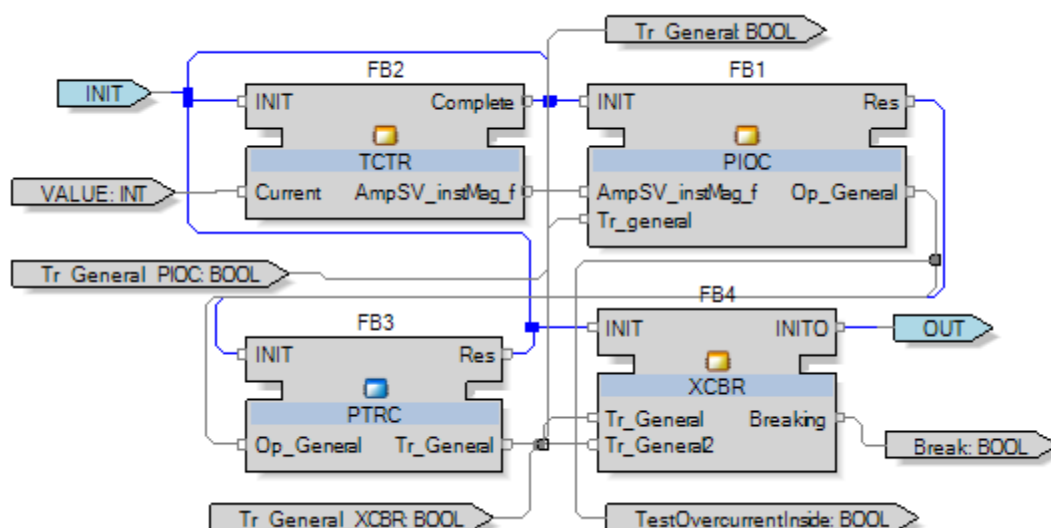


Рис. 2. Реализация одного элемента защиты на ФБ.

На рисунке 3 слева раскрыто содержимое блока *PTCR*. Соответствующая сеть ФБ включает блок *PTRC_Base* и блок временной задержки *E_DELAY* для измерения временных интервалов. Диаграмма управления выполнением (ЕСС) блока *PTRC_Base* приведена на рис. 3 справа.

Рассмотрим работу блока *PTRC_Base*. При поступлении сигнала на событийный вход *INIT* происходит проверка значения на информационном входе *Op_General*. Если оно равно *FALSE*, то происходит переход в состояние *S4*, в котором выдаётся выходной событийный сигнал *Out*, а на выход *Tr_generate* подаётся сигнал *Op_General* (который в данный момент равен *FALSE*). Далее происходит ожидание сигнала *INIT* с *Op_General*, равным *TRUE*, после чего происходит переход в состояние *S1*. Если оно равно *TRUE*, то происходит переход в состояние *S1*. В состоянии *S1* происходит подача сигнала на внешний таймер (выходной сигнал *Start_Time*), в котором происходит отсчёт времени.

Далее возможно два варианта развития событий. Если от таймера пришло событие *Time_end*, то на выход подаётся сигнал *Out* вместе с *Tr_generate*, который равен *Op_General* (который равен *TRUE*, или, если сигнал превышения напряжения внезапно пропал, то *FALSE*). Если в блок пришло событие *INIT* вместе с *Op_General*, равным *FALSE* (это обозначает, что напряжение вернулось в норму), то происходит переход в состояние *S2*, в котором подаётся сигнал останова на внешний таймер (событийный сигнал *Stop_Time*). Независимо от того, дошёл отсчёт таймера до конца или был прерван, происходит переход в начальное состояние.

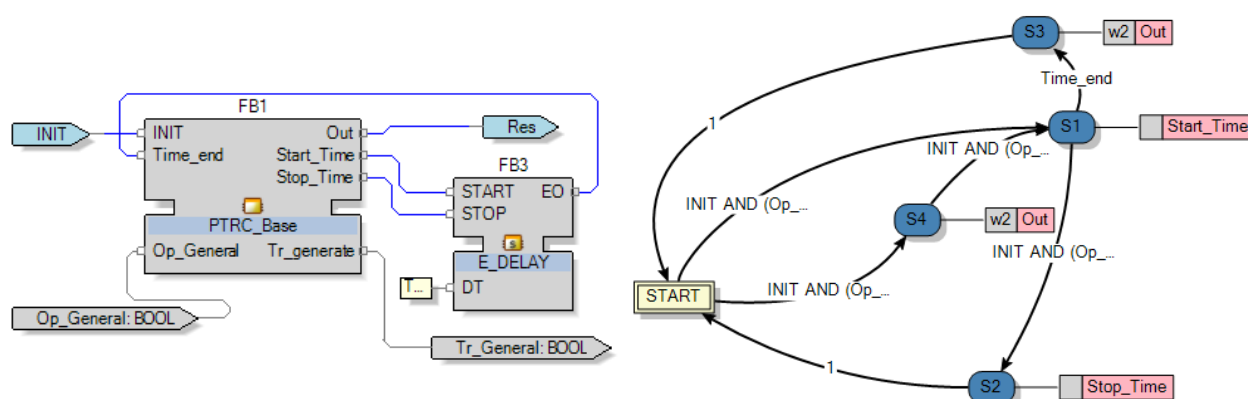


Рис. 3. Содержимое ФБ *PTRC* (слева) и диаграмма ЕСС ФБ *PTRC_Base* (справа)

На рисунке 4 приведен скриншот человеко-машинного интерфейса (ЧМИ) имитационной модели распределенной системы защиты от перегрузок для распределительной сети, представленной на рисунке 1. Как видно из рисунка 4, отображается состояние каждого из четырех *PIED*, входящих в систему. Следует заметить, что для построения ЧМИ использовались САТ-блоки системы *NxtStudio*.

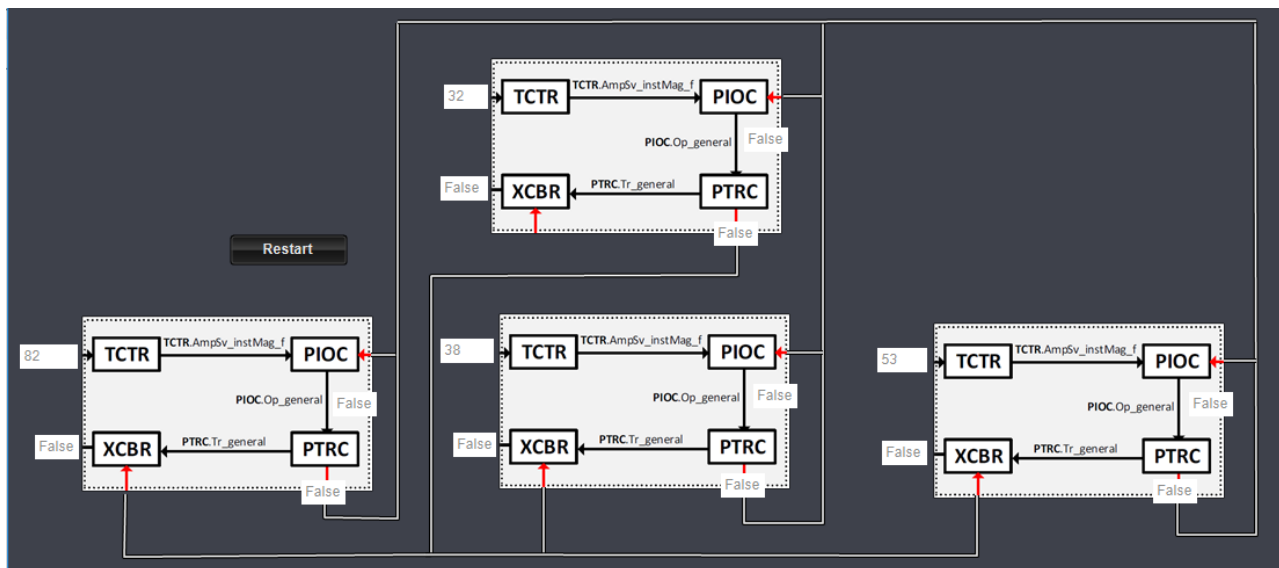


Рис. 4. Человеко-машинный интерфейс имитационной модели распределенной системы защиты от перегрузок для распределительной сети на рисунке 1 в системе NxtStudio

Для автоматизации генерации функционально-блочной модели системы защиты от перегрузок для ЦПС предлагается технологическая цепочка проектирования, приведенная на рис. 5. В отличие от [8] эта цепочка упрощена, поскольку в ней отсутствует онтологическое представление системы и преобразования, связанные с ним. В качестве внутреннего рабочего представления используется язык *Prolog*. Как видно из рис. 5 исходное описание ЦПС представлено на языке *SCL*. В дальнейшем это описание транслируется в *Prolog*-базированное описание ЦПС (уже включающее схему защиты), которое, в свою очередь, преобразуется в описание управляющих ФБ на языке *Prolog*. Для преобразования используется набор правил, также написанных на языке *Prolog*.

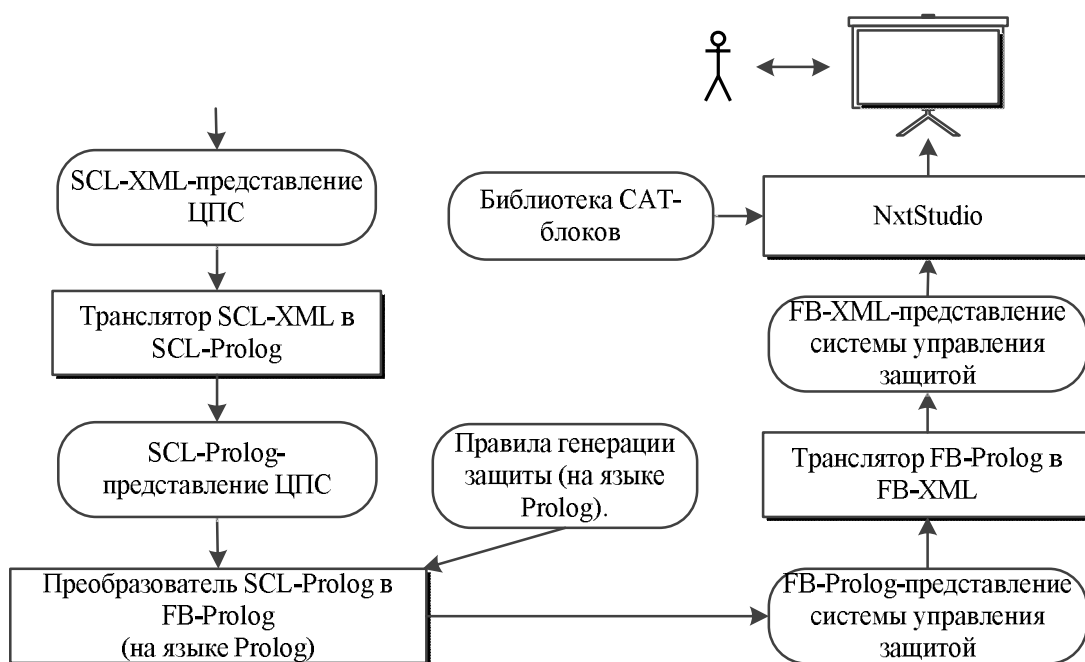


Рис. 5. Структура программного обеспечения для генерации функционально-блочных моделей защиты от перегрузок

Направлением дальнейших исследований является программная реализация системы генерации, приведенной на рисунке 5, а также интеграция системы генерации защиты и системы генерации ЧМИ для автоматизации цифровых подстанций МЭК 61850, описанной в работе [9].

Библиографический список

1. Васильев Д. А., Иващенко В. А., Лукьянов Д. В., Шабельникова А. Ю. Выбор структуры электрических сетей промышленных предприятий в условиях автоматизированного управления электропотреблением // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2010. – № 2 (14). – С. 52–61.
2. Ipakchi A., Albuyeh F. Grid of the future // IEEE Power and Energy Magazine, vol. 7, no. 2, 2009. – P. 52-62.
3. IEC 61850 - Communication networks and systems for power utility automation, IEC Standard, Part 6: Configuration language for communication in electrical substations related to IEDs - Ed 2, 2009.
4. Apostolov A., Vandiver B. IEC 61850 GOOSE Applications to Distribution Protection Schemes // Energize - December 2010. – P. 26-29.
5. Yang C.-W., Vyatkin V. Modeling of IEC 61850 Message Passing for Automatic Generation of Distributed Control // 41st Annual Conference of the IEEE Industrial Electronics Society (IECON 2015), p. 1-6.
6. Vyatkin V. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design, Third Edition. – Instrumentation Society of America (ISA), 2016. – 261 p.
7. Web-site nxtStudio (nxtControl). Available at: <http://www.nxtcontrol.com/>
8. Yang C.-W., Dubinin V., Vyatkin V. Ontology Driven Approach to Generate Distributed Automation Control from Substation Automation Design // IEEE Transactions on Industrial Informatics, 2017, Volume 13, Issue 2, P. 668-679.
9. Voinov A.C., Chen-Wei Yang C.-W., Vyatkin V. Automatic Generation of Function Block Systems Implementing HMI for Energy Distribution Automation // IEEE 15th Int. Conf. of Industrial Informatics (INDIN'2017), Emden, Germany, 24-26 July 2017.

ВНЕДРЕНИЕ АСУ В СФЕРУ ЗДРАВООХРАНЕНИЯ РЕСПУБЛИКИ ТАДЖИКИСТАН

П.М. Шоназаров*, Ф.Т. Холов**

* Южно-Уральский государственный университет,
Россия, Челябинск, shohin900@mail.ru

** Томский государственный университет систем управления и радиоэлектроники,
Россия, Томск, fozil_1990-90@mail.ru.

Аннотация: Данная статья посвящена развитию и внедрению автоматизированных систем в учреждениях здравоохранения Республики Таджикистан. Этот документ будет осуществляться в форме электронных форм обработки документов. Принятие пациентов для начала диагностики и лечения врачей, а также принятие анализов, крови, УЗИ, ЭКГ, хирургическое лечение, прием любых платных услуг, доступных в учреждениях, а также обработка и печать документов пациента в автоматическом режиме. Внедрение автоматизированных систем в сфере здравоохранения было очень успешным. Эта система работает с 2014 года в медицинских учреждениях Республики Таджикистан.

Ключевые слова: Автоматизация, автоматика, управление, безопасность, обработка, технология, программирование, база данных, информация, система, здравоохранение, медицина.

INTRODUCTION OF ASM IN THE HEALTHCARE INSTITUTIONS REPUBLIC OF TAJIKISTAN

P.M. Shonazarov*, F.T. Kholov**

* South Ural State University,
Russia, Chelyabinsk, shohin900@mail.ru

** Tomsk State University of Control Systems and Radioelectronics,
Russia, Tomsk, fozil_1990-90@mail.ru.

Abstract. This article is devoted to the development and implementation of automated systems in healthcare institutions of the Republic of Tajikistan. This document will be implemented in the

form of electronic forms of processing documents. Acceptance of patients for the beginning of diagnosis and treatment of doctors, as well as the acceptance of analyzes, blood, UZI, ECG, surgical treatment, the reception of any fee-based services available in institutions, and the processing and printing of patient records in automatic mode. The introduction of automated systems in the healthcare institutions was very successful. This system has been working since 2014 in medical institutions of the Republic of Tajikistan.

Keywords. Automation, management, safety, processing, technology, programming, database, information, system, diagram, healthcare, medicine.

Введение

Информационные технологии (ИТ) в современном мире широко используются во всех отраслях и также в области здравоохранения. Развитие информационных технологий в области здравоохранения позволяет улучшить обслуживание больных. Многие страны активно используют новые информационные технологии в области здравоохранения. Внедрение информационных технологий в отрасль здравоохранения позволит удаленно проводить консультации для пациентов и работников, обмениваться информацией о пациентах с различными учреждениями и управлять операциями. Все это поднимает отрасль здравоохранения на новый уровень развития технологии ее структур и видов деятельности. Компания ООО «ИЦПА» предлагает новую автоматизированную программу в сфере здравоохранения, которая способствует развитию этой отрасли. Разработка данной системы осуществлялась с использованием [1, 2, 3].

Современные технологии в сфере здравоохранении

Внедрение автоматизации в сфере здравоохранения может улучшить качество услуг, ускорить работу персонала и снизить затраты пациентов. Эти возможности будут доступны в каждой клинике. Предоставление такого современного программного продукта дает каждому своему пользователю огромные возможности.

Информационные технологии в сфере здравоохранения дают возможность решить такие проблемы:

- 1) регистрация больничных пациентов;
- 2) мониторинг и удаленный доступ к информации пациента;
- 3) запись и отправка результата диагностических тестов пациента;
- 4) надлежащий контроль и лечение к назначениям;
- 5) проведение операций пациента.

Медицинские осмотры включают:

- 1) анализ крови;
- 2) анализ сахара в крови;
- 3) биохимический анализ крови;
- 4) общий анализ крови;
- 5) анализ RV;
- 6) УЗИ;
- 7) рентген;
- 8) доплер;
- 9) эхокардиограмма;
- 10) ЭКГ;
- 11) МРТ.

Эта система также имеет функцию контроля информации. В целом, главврач больницы может непосредственно управлять курсом лечения пациента. Информационные технологии позволяют осуществлять высококачественный мониторинг. Учет записи информации в электронном виде во много раз экономит время сотрудников больниц. Вся информация па-

циентов сохраняется в электронном документе, который доступен медицинскому персоналу. Все данные о результатах анализа также регистрируются непосредственно в электронной медицинской карте. Это позволяет профессионалам оценить качество лечения и выявлять диагностические ошибки. Эта программа состоит из следующих интерфейсов.

Интерфейс входа в программу представлен на рисунке 1.

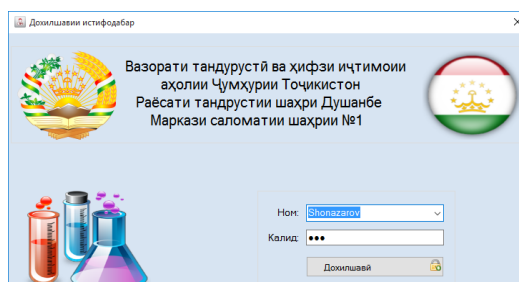


Рис. 1. Вход в программу

Интерфейс регистрации и приёма больных представлен на рис. 2.

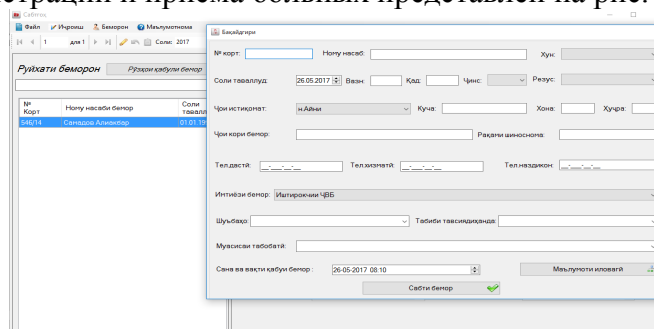


Рис. 2. Интерфейс регистрации больных

В этом интерфейсе приведен прием пациентов. Раньше принимали и регистрировали пациентов на бумагах, теперь те же документы и те же реквизиты используются для регистрации в электронном виде. Эта система, то есть прием пациента, позволяет одному пациенту регистрироваться один раз в повторяющиеся моменты, которые относятся к пациенту, просто зарегистрироваться день в приема. Возможность поиска информации по пациенту представлена на рисунке 3.

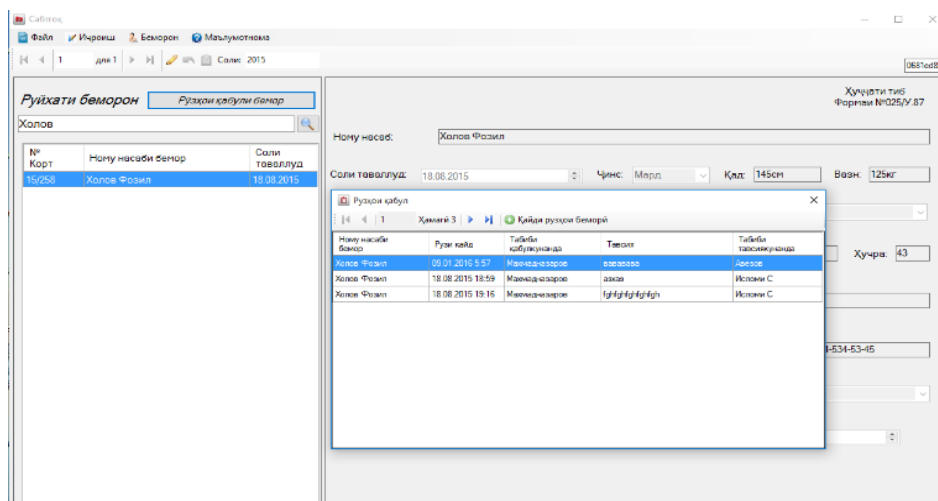


Рис. 3. Интерфейс поиска больных

В этом интерфейсе приведен интерфейс врача, который непосредственно лечит пациента (рис. 4).

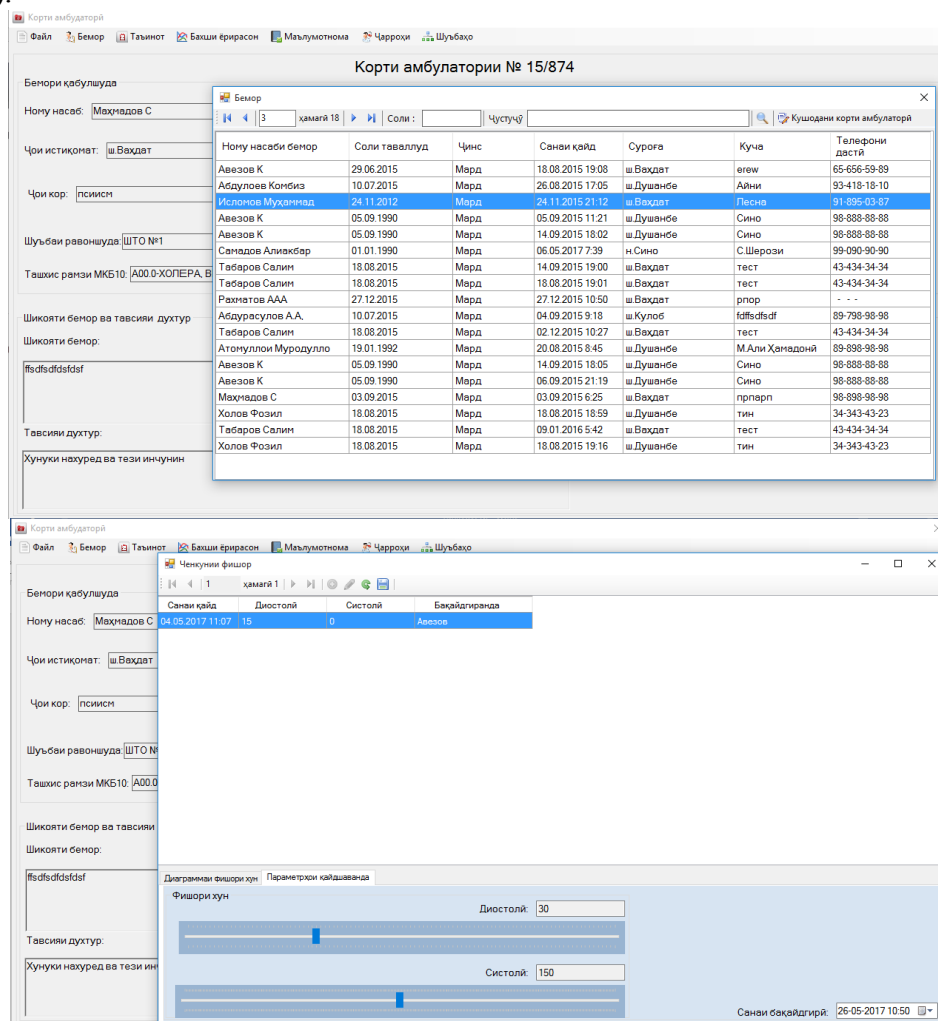


Рис. 4. Интерфейс врача

На следующем рисунке приведен интерфейс лаборатории (рис. 5).

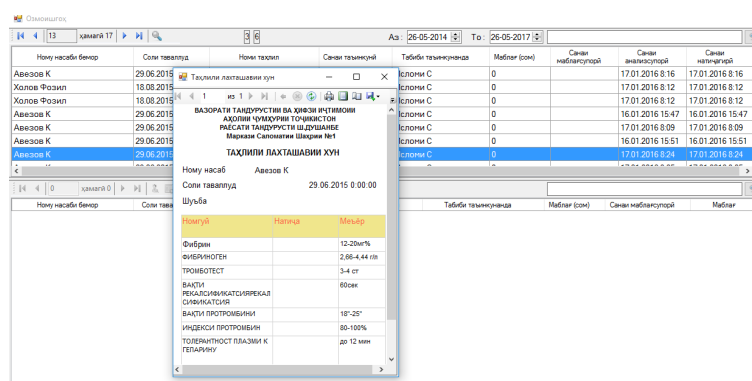


Рис. 5. Интерфейс лаборатории

На этом рисунке приведен интерфейс кассы (рис. 6).

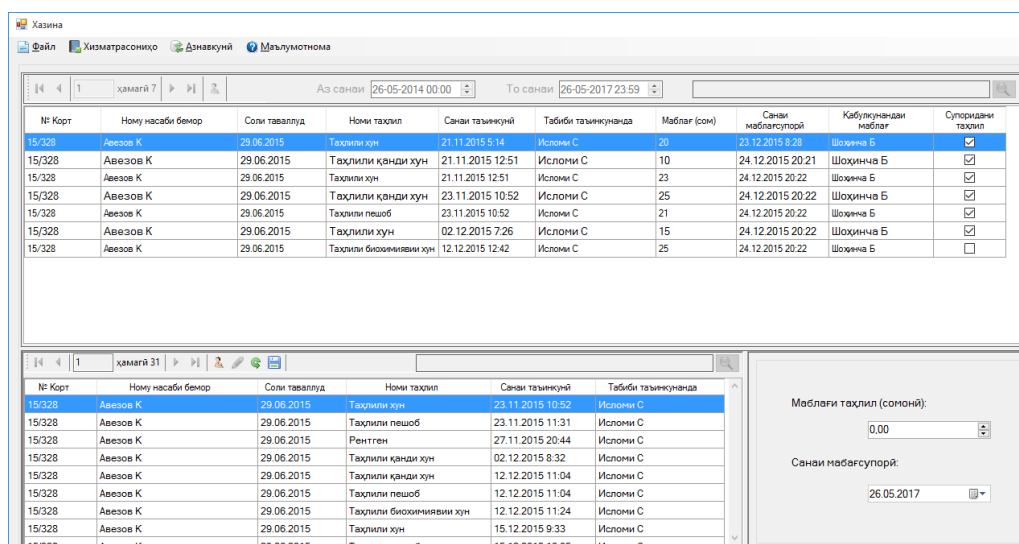


Рис. 6. Интерфейс кассы

На этом рисунке приведен интерфейс управления, которым могут управлять главный врач больницы и лицо, непосредственно ответственное за контроль работы работников своего учреждения. Здесь главный врач может наблюдать выполнения всех действий пациента, такие как оплата за лечение, проведенные анализы, степень лечения и курс лечения (рис. 7).

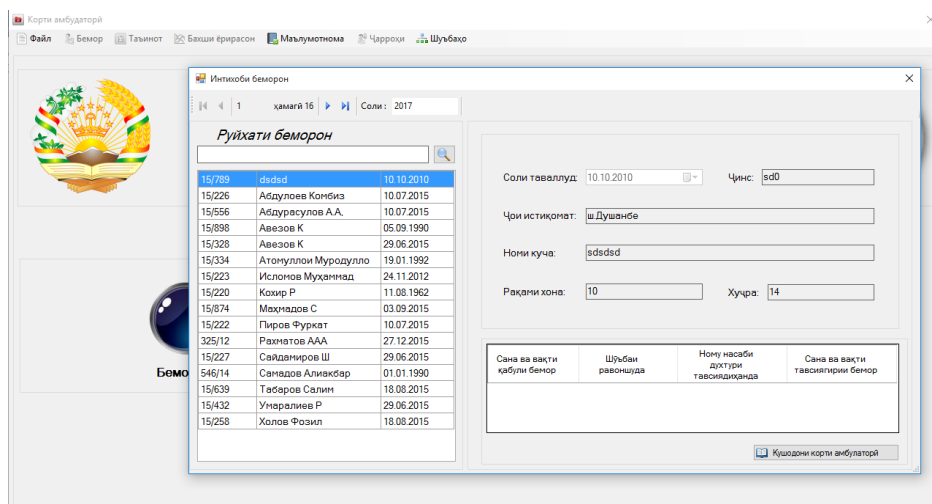


Рис. 7. Интерфейс управления.

Заклучение

В заключении можно сказать, что использование автоматизированных систем во всех структурах дает большую эффективность, снижает коррупцию, экономит время, экономит избыточные затраты и позволяет напрямую управлять промежуточными данными.

Ниже приведен программный код:

```
namespace Ambulatory
{
    public void Imtiyoz()
    {
        try
        {
            Imtiyoz_cmb.Text = "";
            DataSet set2 = new DataSet();
            set2.Clear();
            SqlCommand command02 = new SqlCommand("Select * from Imtiyoz order by
Id", con);
            SqlDataAdapter da1 = new SqlDataAdapter(command02);
            da1.Fill(set2, "Imtiyoz");
            Imtiyoz_cmb.DataSource = set2.Tables["Imtiyoz"];
            Imtiyoz_cmb.DisplayMember = "NomiImtiyoz";
            Imtiyoz_cmb.ValueMember = "Id";
            Imtiyoz_cmb.Text = "";
        }
        catch (Exception x)
        {
            string mess = x.Message;
            MessageBox.Show("mess");
        }
    }
    public void Shahrho()
    {
        try
        {
            DataSet set2 = new DataSet();
            set2.Clear();
            SqlCommand command02 = new SqlCommand("Select * from Shahr order by
NomiShahr", con);
            SqlDataAdapter da1 = new SqlDataAdapter(command02);
            da1.Fill(set2, "Shahr");
        }
    }
}
```

```
        Shahr_cmb.DataSource = set2.Tables["Shahr"];
        Shahr_cmb.DisplayMember = "NomiShahr";
        Shahr_cmb.ValueMember = "IdShahr";
        Shahr_cmb.Text = "";
    }
    catch (Exception x)
    {
        string mess = x.Message;
        MessageBox.Show("mess");
    }
}

public void Tabibho()
{
    try
    {
        DataSet set2 = new DataSet();
        set2.Clear();
        SqlCommand command02 = new SqlCommand("Select IdUser,FIODukhtur from
Users where IdRol=3 and Idshuba=" + comboBox6.SelectedValue + " order by
FIODukhtur", con);
        SqlDataAdapter da1 = new SqlDataAdapter(command02);
        da1.Fill(set2, "Users");
        comboBox4.DataSource = set2.Tables["Users"];
        comboBox4.DisplayMember = "FIODukhtur";
        comboBox4.ValueMember = "IdUser";
        comboBox4.Text = "";
    }
}
```

Библиографический список

1. Ричард У. Java 8 Lambdas Functional Programming for the Masses / Ричард У. // У. Влиссидес Дж, Издательство: ДМК Пресс Год: 2014 Язык: Русский ISBN: 978-5-94074-919-6 Страниц: 194
2. Гамма Э. Приемы объектно-ориентированного проектирования. / Гамма Э. // Джонсон Р. // Влиссидес Дж. Паттерны проектирования. - СПб.: Питер, 2001. - 368 с.
3. Роберт Лафоре. Структуры данных и алгоритмы JAVA. / Роберт Лафоре Издательство: Питер Год: 2013 Язык: Русский ISBN: 978-5-496-00740-5 Страниц: 704

УДК 697.341; ГРНТИ 44.01.85

ИДЕНТИФИКАЦИЯ МОДЕЛЕЙ ПРИ ПРОЕКТИРОВАНИИ ИНЖЕНЕРНЫХ СИСТЕМ ЗДАНИЙ

О.Ю. Марьясин, А.А. Огарков

*Ярославский государственный технический университет,
Россия, Ярославль, maryasinou@ystu.ru*

Аннотация. Рассмотрены возможности совместного моделирования ВЕМ системы EnergyPlus и пакета MATLAB для получения моделей объектов управления, в форме традиционной для теории автоматического управления. Полученные модели затем можно использовать для реализации оптимальных и адаптивных законов управления микроклиматом и энергопотреблением зданий.

Ключевые слова. HVAC, EnergyPlus, идентификация, MLE+, MATLAB, Simulink

IDENTIFICATION OF MODELS AT DESIGNING ENGINEERING SYSTEMS OF BUILDINGS

O.Yu Maryasin, A.A Ogarkov

Yaroslavl State Technical University,

Russia, Yaroslavl, maryasinou@ystu.ru

Abstract. This paper considered the possibilities co-simulation of the EnergyPlus system and the MATLAB package for obtaining models of control objects in the form of the traditional for the theory of automatic control. The resulting models can be used to implement optimal and adaptive control laws for quality management of the indoor environment in buildings with energy saving.

Keywords. HVAC, EnergyPlus, identification, MLE+, MATLAB, Simulink

Введение

В последнее время при проектировании зданий все шире используются технологии информационного моделирования зданий (BIM - Building Information Modeling) и энергетического моделирования зданий (BEM - Building Energy Modeling). Они позволяют существенно снизить затраты на проектирование зданий, обеспечивают выполнение требований по энергосбережению и повышению энергетической эффективности, гарантируют комфортную и здоровую среду в помещениях здания. Данные о конструкции и материалах здания из BIM, используются для проведения энергомоделирования в BEM, а построенные там модели используются при проектировании систем отопления, вентиляции и кондиционирования (ОВК или HVAC, как их называют на западе).

Для энергомоделирования, расчетов и моделирования HVAC систем чаще всего используются специализированные программные пакеты, такие как Trane TRACE 700, EnergyPlus, ESP-r, IDA ICE, TRNSYS, HVACSIM+ и другие [1]. Эти BEM системы выполняют моделирование на основе системы дифференциальных и/или алгебраических уравнений, определяющих теплофизические характеристики зданий и HVAC систем, в том числе, с заданными стратегиями эксплуатации и управления. При этом моделирование HVAC систем, как правило, включает и моделирование систем управления инженерным оборудованием.

Уровень моделирования систем управления в различных инструментах может отличаться. Системы управления могут быть представлены в виде моделей диспетчерского управления, определяющих профили отопительных и холодильных нагрузок, как например, в EnergyPlus или как модели локального управления, как в ESP-r и TRNSYS. Более сложные и продвинутое модели контроллеров, такие как оптимальные, адаптивные, контроллеры на базе нечеткой логики, нейроконтроллеры и др., доступны только в инструментах на базе общих пакетов для научных и инженерных расчетов, таких как MATLAB, SCILAB, SimInTech, пакетах, основанных на языке Modelica (Dymola, OpenModelica, JModelica) или в инструментах, взаимодействующих с указанными пакетами (TRNSYS).

Для реализации оптимальных и адаптивных законов управления необходимо наличие моделей объектов управления. В качестве таких моделей, чаще всего используются модели в виде передаточных функций или уравнений состояния. BEM-системы пока не позволяют формировать такие модели. Получить их экспериментальным путем на стадии проектирования тоже не представляется возможным. Поэтому, обычно их получают путем создания компьютерных моделей, в упомянутых ранее пакетах для научных и инженерных расчетов, на основе известных аналитических или экспериментально-аналитических зависимостей. Построение таких моделей требует дополнительных трудозатрат высококвалифицированных специалистов. При этом такие модели в большинстве случаев дают гораздо менее точные и более приближенные результаты, чем полученные по модели BEM системы.

Инструменты для совместного моделирования

Авторы предлагают использовать для получения моделей объектов управления, технологии совместного моделирования BEM систем и пакетов для научных и инженерных расчетов. Так, например, с помощью средства Building Controls Virtual Test Bed (BCVTB) [2] или библиотеки MLE+ [3] можно организовать совместное моделирование MATLAB/Simulink с BEM системой EnergyPlus. То есть BEM систему, в которой уже создана модель микроклимата и энергопотребления здания, адекватно описывающая реальный объект, предлагается использовать в качестве “экспериментального” объекта для построения моделей объектов управления.

Так как EnergyPlus фактически представляет собой расчетный движок и не имеет собственного графического интерфейса, то для взаимодействия пользователя с этой BEM системой используются дополнительные графические приложения, такие как DesignBuilder, Simergy, OpenStudio и др. Эти приложения-оболочки, с помощью открытых форматов IFC и gbXML, позволяют импортировать данные из информационных моделей BIM систем. Окно программы Simergy с загруженной в нее информационной моделью показано на рисунке 1. Для выполнения энергомоделирования загруженную информационную модель, почти всегда необходимо дорабатывать в программе-оболочке. Программа-оболочка позволяет создать входной текстовый IDF файл с данными для системы EnergyPlus. Для выполнения совместного моделирования требуется дальнейшая ручная доработка полученного IDF файла.

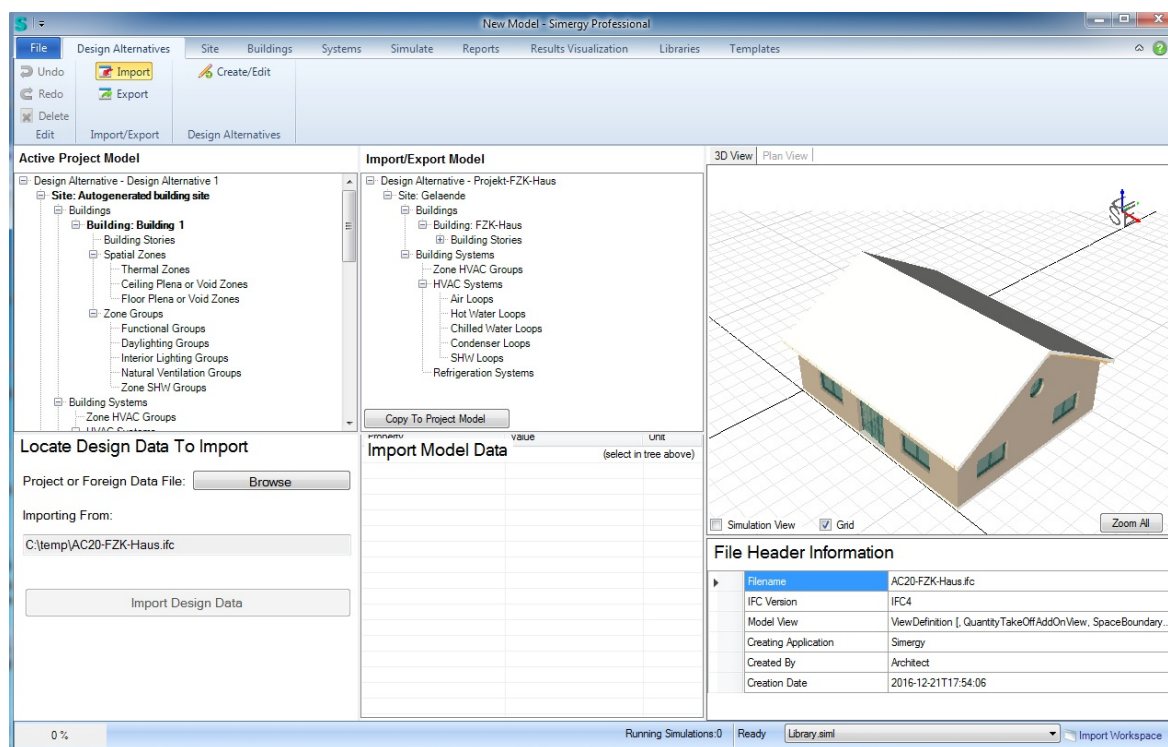


Рис. 1. Вид главного окна программы Simergy

BCVTB является более универсальным инструментом, так как позволяет организовать совместное моделирование не только MATLAB/Simulink с EnergyPlus, но и с другими системами, такими как Dymola, Radiance, ESP-r и TRNSYS. Для взаимодействия между системами BCVTB использует свободно-доступную графическую среду моделирования Ptolemy II. Ptolemy II включает блок Simulator, который позволяет запускать внешнюю среду моделирования, передавать ей входные параметры, получать от нее результаты моделирования и транслировать их в выходной порт. Взаимодействие между различными процессами реализуется с

использованием Berkeley Software Distribution socket interface (BSD sockets). Ниже показана структура файла, описывающего переменные, используемые для обмена информацией с системой EnergyPlus:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<BCVTB-variables>
  <variable source="EnergyPlus">
    <EnergyPlus name="ENVIRONMENT" type="Site Outdoor Air Temperature"/>
  </variable>
  <variable source="EnergyPlus">
    <EnergyPlus name="ZSF1" type="Zone Air Temperature"/>
  </variable>
  <variable source="Ptolemy">
    <EnergyPlus schedule="TSetHea"/>
  </variable>
  <variable source="Ptolemy">
    <EnergyPlus schedule="TSetCoo"/>
  </variable>
</BCVTB-variables>
```

Входные переменные EnergyPlus должны быть описаны в секциях ExternalInterface: Schedule, ExternalInterface: Actuator and ExternalInterface: Variable входного IDF файла. Выходные переменные – в секциях Output: Variable и EnergyManagementSystem: OutputVariable.

Библиотеки MLE+ в отличие от BCVTB позволяет организовать совместное моделирование MATLAB/Simulink только с системами EnergyPlus и Radiance. В MLE+ применяются некоторые механизмы BCVTB, но не используется промежуточная среда моделирования Ptolemy II. MLE+ включает удобную графическую утилиту, которая позволяет настраивать интерфейс с переменными EnergyPlus для решения задач идентификации и проектирования систем управления, запускать и отображать результаты моделирования, а также формировать вспомогательные файлы, необходимые для совместного моделирования.

Авторы опробовали оба варианта организации совместного моделирования MATLAB/Simulink с BEM системой EnergyPlus. BCVTB работает с MATLAB/Simulink в режиме сервера автоматизации, что исключает интерактивное взаимодействие пользователя с MATLAB/Simulink во время моделирования. Учитывая это и другие обстоятельства, авторы остановились на использовании библиотеки MLE+.

Идентификация модели из системы EnergyPlus

Блок-диаграмма Simulink, использованная для идентификации модели из BEM системы EnergyPlus, показана на рисунке 2.

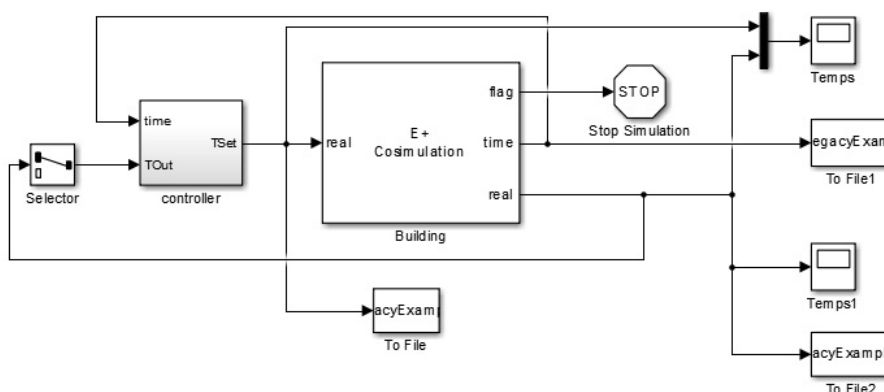


Рис. 2. Блок-диаграмма Simulink

Основным блоком здесь является блок E+ Cosimulation, с помощью которого осуществляется взаимодействие с системой EnergyPlus. В качестве параметров блока задается путь к исполняемым файлам EnergyPlus, имя IDF файла, имя файла с погодными данными и другие. Блок controller предназначен для реализации модели системы управления микроклиматом здания. При проведении идентификации этот блок использовался для формирования необходимых входных сигналов для системы EnergyPlus. Группа блоков To File позволяет сохранить “экспериментальные” данные, необходимые для проведения идентификации.

Пакет MATLAB предлагает широкий набор инструментов для решения задач идентификации различных объектов. Однако, авторы воспользовались тем фактом, что переходные процессы, описывающие динамику теплового режима зданий, имеют вид близкий к апериодическому [4]. Это позволяет нам заранее ограничить вид используемых моделей апериодическими передаточными функциями и применить более простую процедуру идентификации. Поэтому для проведения идентификации мы воспользуемся функцией `lsqcurvefit` Optimization toolbox пакета MATLAB. Эта функция реализует алгоритм подгонки кривой с использованием нелинейного метода наименьших квадратов.

Результаты расчетов за период времени около трех часов представлены на рисунке 3. На рисунке 3 сплошной линией показаны “экспериментальные” значения температуры в помещении, штрих-пунктирной линией показано изменение температуры внешней среды, а пунктирной линией, значения температуры в помещении, рассчитанные по модели, полученной в результате идентификации. Из рисунка видно, что на рассматриваемом промежутке времени погрешность вычисления температуры составляет не более 2 – 3 °С. Такой погрешности вполне достаточно для использования полученной модели, например, в алгоритмах оптимального прогнозирующего управления – Model Predictive Control (MPC).

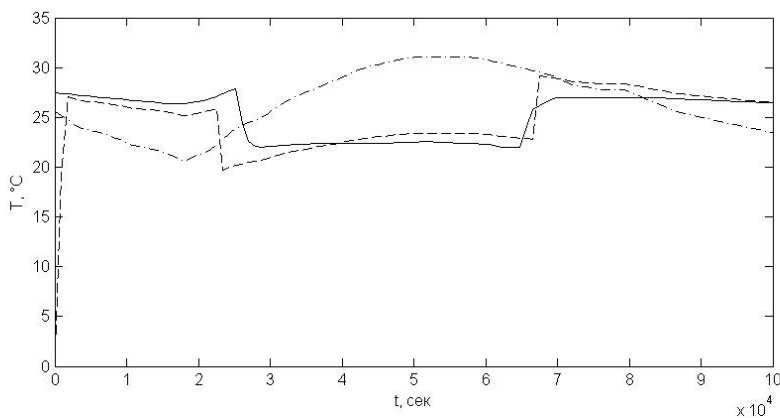


Рис. 3. Результаты расчетов

Выводы

Таким образом, в результате работы показано, что возможности совместного моделирования ВЕМ систем и пакетов для научных и инженерных расчетов можно использовать для получения моделей объектов управления, в форме традиционной для теории автоматического управления. Полученные модели затем можно использовать для реализации оптимальных и адаптивных законов управления микроклиматом и энергопотреблением зданий. Это позволит обеспечить поддержание комфортного микроклимата в зданиях и существенно снизить потребление энергоресурсов.

Библиографический список

1. Марьясин О.Ю., Колодкина А.С. Пакеты компьютерного моделирования HVAC систем и энергопотребления зданий. Математика и естественные науки. Теория и практика: Межвуз. сб. науч. тр. Вып. 11. – Ярославль: Издат. дом. ЯГТУ, 2016. – с. 306 – 315.
2. Building Controls Virtual Test Bed [Электронный ресурс], URL: <https://simulationresearch.lbl.gov/bcvtb> (дата обращения: 07.02.2018).
3. MLE+ [Электронный ресурс], URL: https://github.com/mlab-upenn/mlep_v1.1 (дата обращения: 07.02.2018).
4. Чистович, С.А. Автоматизированные системы теплоснабжения и отопления. – Л.: Стройиздат, 1987. – 248 с.

УДК 004.42

ПРИЛОЖЕНИЕ ДЛЯ КОНФИГУРИРОВАНИЯ КОМПОНЕНТОВ НЕЙРОСЕТЕВЫХ ПРЕОБРАЗОВАТЕЛЕЙ ЧАСТОТНО-ВРЕМЕННЫХ ПАРАМЕТРОВ СИГНАЛОВ

А. Барсуков, А. Антоненко

*Рязанский государственный радиотехнический университет,
Российская федерация, Рязань, dervy@mail.ru*

Аннотация. В данной работе реализовано программное приложение для конфигурирования нейронов-преобразователей. Описана функциональная модель нейрона-преобразователя «двоичный умножитель». Приведен сгенерированный VHDL-код, описывающий нейрона-преобразователя «двоичный умножитель» с выходом в виде широтно-импульсной модуляции.

Ключевые слова. Нейрон, преобразователь, двоичный умножитель, VHDL, частота, код.

APPLICATION FOR CONFIGURING THE COMPONENTS NEURAL NETWORK CONVERTERS OF SIGNALS TIME-FREQUENCY PARAMETERS

A. Barsukov, A. Antonenko

*Ryazan State Radio Engineering University,
Russian Federation, Ryazan, dervy@mail.ru*

Abstract. This paper implemented a software application for configuration of neurons-converters. The functional model of the neuron-converter "binary multiplier" is described. The generated VHDL-code describing the neuron-converter "binary multiplier" with the output in the form of pulse-width module is given.

Keywords. Neuron, converter, binary multiplier, VHDL, frequency, code.

Введение

Основными компонентами преобразователей формы представления информации (ПФИ) основанных на математическом аппарате искусственных нейронных сетей (ИНС-преобразователей) являются нейроны-преобразователи (НП) и аналого-цифровые синаптические связи между ними [1]. Под нейроном-преобразователем понимается математический нейрон с гибридной формой представления информации и возможностью ее преобразования из одного вида в другой с совместным выполнением несложной математической операции.

Целью данной работы являлось решение проблемы конфигурирования НП в зависимости от требуемой разрядности и вариантов включения (загрузок). Решение заключается в разработке программного приложения, осуществляющего конфигурацию компонентов преобразователя (модулей) с заданной разрядностью и вариантом включения. Такая программа позволяет производить требуемые изменения компонентов преобразователя для построения преобразователя в соответствии с решением новой поставленной задачи.

Функциональные модели НП

Функциональную модель НП удобно представить как операционное устройство на основе типовых элементов и узлов вычислительной техники (ВТ). Такой подход позволяет перейти к программной реализации этих моделей на языках описания аппаратуры (HDL) таких как VHDL и Verilog.

Нейроны-преобразователи оперируют гибридными (аналого-цифровыми) сигналами на входах и выходах [2]. Это порождает большое многообразие типов НП, выполняющих операции над сигналами разных форм представления информации. НП определяются следующими характеристиками: формой представления информации на входах и выходах, видом выполняемого преобразования, составом элементов и узлов ВТ, разрядностью базового узла ВТ. Для НП частотно-временных параметров сигналов в качестве базового узла ВТ, как правило выступают счетчики, сумматоры, регистры и т.п.

В качестве примера рассмотрим нейрон-преобразователь “ $N \rightarrow n$ ” и “ $N \rightarrow F$ ” (рис. 1) с выходом в виде количества единичных импульсов $n_y(j2^k T_0)$ за период $T_H = 2^k T_0$ и в виде среднего значения частоты $f_y = f_0 \frac{N_x}{2^k}$.

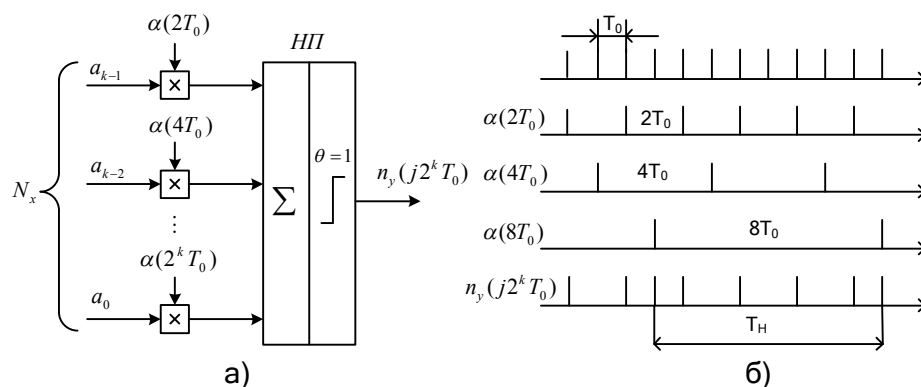


Рис. 1. Нейрон-преобразователь “ $N \rightarrow n$ ” и “ $N \rightarrow F$ ”

НП (см. рис. 1) является устройством непрерывного принципа действия. Входами НП “ $N \rightarrow F$ ”, известного как двоичный умножитель (ДУ), являются следующие переменные:

- $x^* = N_x = a_{k-1}a_{k-2}a_{k-3}...a_0 = \sum_{l=0}^{k-1} a_l \cdot 2^l$ – цифровой k -разрядный двоично-позиционный код, представляющий цифровой эквивалент x^* , каждый l -й бит (разряд) которого подается на отдельный вход сумматора НП, $l = \overline{[0, k-1]}$;
- $\alpha(2T_0), \alpha(4T_0), \dots, \alpha(2^k T_0)$ – набор заданных на интервале $T_H = 2^k T_0$ текущих потоков единичных импульсов $\alpha(2^l T_0)$ в виде числа импульсов 2^l или интенсивности (частоты) их следования $f_0 / 2^l$, значения которых определяются весами 2^l битов $a_l = \{0, 1\}$ кода N_x , где T_0 – период опорной частоты $f_0 = 1/T_0$ как эталона преобразования.

Функционирование нейрона-преобразователя (см. рис. 1) описывается совокупностью операций:

$$y = F_{\text{НП}}(G),$$

$$F_{\text{НП}} = \begin{cases} 1(jT_0), & \text{если } G \geq \theta; \\ 0 & \text{иначе,} \end{cases}$$

$$G = \sum_{l=0}^{k-1} a_l \cdot \alpha(2^{k-l} T_0),$$

где $F_{\text{НП}}$ – импульсная пороговая функция активации НП с порогом $\theta = 1$;
 G – сумма взвешенных входов, которая в развернутом виде описывается выражением:

$$G(iT_0) = a_{k-1} \cdot \alpha(2T_0) + a_{k-2} \cdot \alpha(4T_0) + a_{k-3} \cdot \alpha(8T_0) + \dots + a_0 \cdot \alpha(2^k T_0).$$

Выходная переменная y нейрона-преобразователя “ $N \rightarrow n$ ” представлена в виде числа (количества) $n_y(j2^k T_0)$ импульсов за интервал $T_H = 2^k T_0$, формируемых как сумма отдельных потоков (чисел) $\alpha(2T_0), \alpha(4T_0), \dots, \alpha(2^k T_0)$ в соответствии со значениями разрядов $a_l = \{0, 1\}$ кода $N_x = a_{k-1}a_{k-2}a_{k-3} \dots a_0$:

$$n_y = a_{k-1} \cdot \alpha(2T_0) + a_{k-2} \cdot \alpha(4T_0) + a_{k-3} \cdot \alpha(8T_0) + \dots + a_0 \cdot \alpha(2^k T_0)$$

или

$$n_y = a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + \dots + a_12^1 + a_02^0$$

с учетом того, что $\alpha(2^l T_0) = 2^{k-l}$. Операция умножения двоичной переменной a_l на значение $\alpha(2^l T_0)$ осуществляется на основе логической схемы «И», а суммирование – с помощью дизъюнктора «ИЛИ».

Поскольку $N_x = a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + a_{k-3}2^{k-3} + \dots + a_02^0$, то формируемый на выходе НП “ $N \rightarrow n$ ” результат будет представлен как:

$$n_y = N_x. \quad (1)$$

Поделив левую и правую части выражения (1) на значение $T_H = 2^k T_0$, можно определить среднее значение частоты $f_y = \frac{N_x}{T_0 \cdot 2^k}$ или $f_y = f_0 \frac{N_x}{2^k}$ следования импульсов на выходе НП, как число импульсов за интервал T_H :

$$f_y = n_y / 2^k T_0 = \frac{a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + a_{k-3}2^{k-3} + \dots + a_02^0}{2^k T_0}.$$

С учетом того, что $n_y = N_x$, получим описание операции преобразования НП “ $N \rightarrow F$ ” в виде среднего значения частоты следования импульсов, линейно зависящего от преобразуемого кода:

$$f_y = f_0 \frac{N_x}{2^k}$$

за интервал $T_H = 2^k T_0$.

Программное приложение для конфигурирования НП

Разрабатываемое программное приложение целесообразно рассматривать как часть более крупного проекта, осуществляющего синтез ИНС-преобразователей на основе НП, полученных с помощью разрабатываемого в данной работе приложения. Такой подход к проектированию функциональных преобразователей ПФИ позволяет формализовать проектные процедуры, что приведёт к значительному уменьшению времени разработки конечного устройства.

В приложении пользователю предоставляется выбор программной конфигурации НП в зависимости от выполняемых им функций математического преобразования и разрядности цифрового кода. Также добавлена возможность выбора имени генерируемого программного файла и каталога для его расположения (рис. 2).

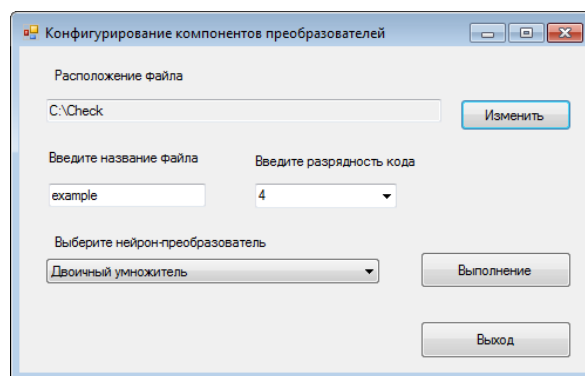


Рис. 2. Внешний вид программного приложения

Также доступны различные варианты начальных загрузок, связанных с особенностями внутренней организации (рис. 3).

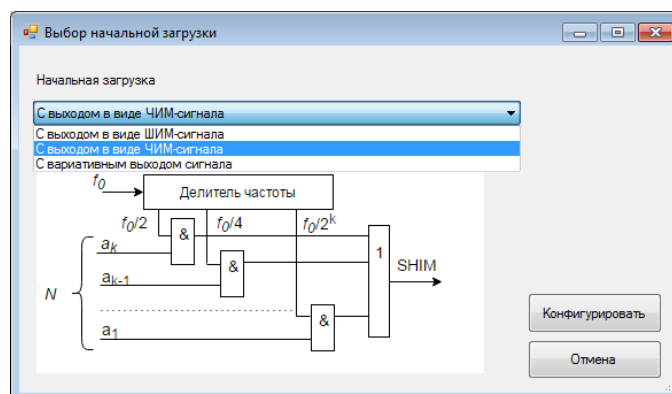


Рис. 3. Варианты начальных загрузок для НП «двоичный умножитель»

Далее приведен VHDL-код сгенерированный приложением. Код описывает НП «двоичный умножитель» с выходом в виде широтно-импульсной модуляции:

```
library ieee;
library altera_mf;
```

```
use ieee.std_logic_1164.all;
use altera_mf.altera_mf_components.all;
use ieee.std_logic_unsigned.all;

entity vhdl_du is
  port(f0 :in std_logic; -- опорная частота
    w : in std_logic_vector (4 downto 1); -- весовой коэффициент
    f_2_o, f_4_o, f_8_o, f_16_o : out std_logic; -- разделенная частота
    shim : out std_logic -- выход в виде широтно-импульсной модуляции
  );
end vhdl_du;

architecture behav of vhdl_du is
  signal cnt : std_logic_vector(4 downto 1);
  signal cnt_t : std_logic_vector(4 downto 1);
  signal f_2, f_4, f_8, f_16 : std_logic;
  signal shim_in : std_logic;

begin

  process(f0)
  begin
    if (f0='1' and f0'event)
      then cnt <= cnt + 1;
    end if;
  end process;

  process(f0)
  begin
    if (f0='1' and f0'event)
      then cnt_t <= cnt;
    end if;
  end process;

  f_2 <= w(4) and cnt(1) and (not cnt_t(1));
  f_4 <= w(3) and cnt(2) and (not cnt_t(2));
  f_8 <= w(2) and cnt(3) and (not cnt_t(3));
  f_16 <= w(1) and cnt(4) and (not cnt_t(4));

  shim_in <= f_2 or f_4 or f_8 or f_16;

  f_2_o <= f_2;
  f_4_o <= f_4;
  f_8_o <= f_8;
  f_16_o <= f_16;
  shim <= shim_in;

end behav;
```

Выводы

Программное приложение (свидетельство о государственной регистрации программы для ЭВМ № 2017661231), осуществляющее автоматическое генерирование программного кода на основе предустановленных пользователем параметров, является удобным и функциональным решением поставленной задачи. Такой подход к программной реализации НП позволяет широко использовать способность ПЛИС к реконфигурированию, что увеличивает возможный спектр решаемых задач.

Работа выполнена при поддержке российского фонда фундаментальных исследований (Проект № 16-38-00388 мол_а).

Библиографический список

1. Локтюхин В.Н., Челебаев С.В. Нейросетевые преобразователи импульсно-аналоговой информации: организация, синтез, реализация // Под общей редакцией А.И. Галушкина. – М.: Горячая линия – Телеком, 2008. – 144 с.
2. Локтюхин В.Н., Челебаев С.В., Антоненко А.В. Нейросетевые преобразователи информации: синтез и программирование на ПЛИС. – Рязань: Изд-во «Сервис», 2011. – 128 с.

УДК 004.023

АНАЛИЗ ЭКСПЕРТНЫХ ОЦЕНОК ПРОЕКТОВ НА ОСНОВЕ КОЭФФИЦИЕНТОВ РАНГОВОЙ КОРРЕЛЯЦИИ

С.В. Скворцов, В.И. Хрюкин

Рязанский государственный радиотехнический университет,

Российская Федерация, Рязань, vi_xr@mail.ru

Аннотация. Рассмотрены вопросы практического применения методов экспертных оценок программных проектов, основанных на корреляционном анализе. Показана ограниченность использования таких методов в случае несогласованности оценок экспертов. Предложен алгоритмический подход к определению групп экспертов, мнения которых согласованы, основанный на использовании группового коэффициента ранговой корреляции.

Ключевые слова. Экспертные оценки, коэффициент корреляции Спирмена, коэффициент конкордации, групповой коэффициент ранговой корреляции

ANALYSIS OF EXPERT ESTIMATES OF PROJECTS BASED ON RANK CORRELATION COEFFICIENTS

S. V. Skvortsov, V. I. Khryukin

Ryazan State Radio Engineering University,

Russian Federation, Ryazan, vi_xr@mail.ru

Abstract. The questions of practical application of methods of expert estimates of software projects based on correlation analysis are considered. Limited use of such methods in case of inconsistency of expert opinions is shown. An algorithmic approach for the definition of groups of experts whose opinions are consistent, based on the use of the group rank correlation coefficient is proposed.

Keywords. Expert estimates, the Spearman correlation coefficient, the coefficient of concordance, cluster analysis, group rank correlation coefficient

Введение

При экспертных оценках качества предлагаемых программных проектов возникает необходимость установить тесноту связи между порядковыми (ординарными) переменными. В этом случае объекты анализа (проекты) каждым экспертом упорядочиваются (ранжируются) по степени важности оцениваемого параметра. Для этого каждому объекту присваивается определенный номер, называемый рангом. Если n объектов ранжированы двумя экспертами,

то имеется возможность оценить тесноту связи между переменными, основываясь на рангах с использованием коэффициента корреляции Спирмена [1].

В случае анализа значимых проектов изначально привлекают большее количество специалистов. Тогда возникает необходимость оценить качество проектов по данным m экспертов ($m > 2$) и определить мнения каких экспертов согласуются между собой. Согласованность мнений нескольких экспертов определяется с помощью множественного коэффициента корреляции рангов (коэффициента конкордации) [2], который вычисляется по формуле

$$W = \frac{12 \sum_{i=1}^n (\sum_{j=1}^m r_{ij} - \bar{r}_i)^2}{m^2 (n^3 - n)}, \quad (1)$$

где r_{ij} - ранг i -го объекта, определенный j -м экспертом;

$\bar{r}_i$ - оценка математического ожидания i -го объекта.

Проверка значимости коэффициента конкордации основана на распределении хи-квадрат. Она выполняется следующим образом.

1. Рассчитывается величина $\chi^2 = n(m-1)W$.

2. Полученное значение χ^2 сравнивается с критическим значением распределения $\chi^2_{кр}$ для принятого уровня значимости α и числа степеней свободы $f = (n-1)(m-1)$, которое определяется из справочных таблиц [3].

3. Если $\chi^2 > \chi^2_{кр}$, то гипотеза об отсутствии связи отвергается, а корреляция признаётся значимой. В противном случае гипотеза об отсутствии связи принимается.

При наличии связанных (одинаковых) рангов формулы для расчета W и χ^2 существенно усложняются [2]:

$$W = \frac{12 \sum_{i=1}^n (\sum_{j=1}^m r_{ij} - \bar{r})^2}{m^2 (n^3 - n) - m \sum_{j=1}^m T_j}. \quad (2)$$

Здесь $\bar{r}$ - средний ранг, равный $\bar{r} = \frac{1}{n} \sum_{i=1}^n \bar{r}_i$;

T_j - показатель связанных рангов в j -ранжировке, который определяется как

$$T_j = \sum_{k=1}^{H_j} (h_k^3 - h_k), \quad (3)$$

где H_j - число групп равных рангов в j -ой ранжировке;

h_k - число равных рангов в k -ой группе связанных рангов при ранжировке j -м экспертом.

Коэффициент конкордации позволяет определить, случайна или не случайна согласованность мнений специалистов. Чем больше коэффициент конкордации, тем выше степень согласованности мнений специалистов. Коэффициент может принимать значения $0 \leq W \leq 1$. Если $W = 0$, то это означает полное отсутствие согласованности между ранжировками экс-

пертов, а значение $W = 1$ показывает, что специалисты одинаково упорядочили объекты. При $W \geq 0,6$ можно говорить о сильной положительной ранговой связи, что позволяет утверждать о примерно одинаковой оценке проектов экспертами. Величина $W < 0,6$ означает, что эта связь слабая или умеренная, т.е. оценки экспертов связаны друг с другом несущественно.

Использование коэффициента конкордации имеет ограниченное применение, так как только позволяет установить согласованы или нет оценки экспертов. В последнем случае возникает необходимость дополнительных исследований, например, направленных на определение групп специалистов, мнения которых согласованы.

Для этого можно воспользоваться методами кластерного анализа, который позволяет выделить однородные в некотором смысле кластеры, т.е. группы экспертов, оценки объектов у которых в значительной степени совпадают [4].

Постановка задачи

Недостатком последнего подхода является то, что каждый эксперт может быть включен только в один из кластеров (группу). Это исключает возможность получения всех допустимых вариантов распределения экспертов по кластерам, которые могут использоваться для дополнительного анализа проектов. Для решения этой задачи предлагается подход, связанный с нахождением таких групп экспертов, оценки которых в значительной степени совпадают. Мету совпадения мнений специалистов в этих группах предлагается количественно определять с помощью *группового* коэффициента ранговой корреляции, который показывает возможность включения эксперта в некоторую группу, если его значение не меньше, чем 0,6. В этом случае любой эксперт может входить сразу в несколько подобных групп.

В качестве окончательного результата анализа принимается группа экспертов, которая включает в себя наибольшее число специалистов. Если таких групп несколько, то выбирается та группа, для которой итоговый групповой коэффициент ранговой корреляции наибольший. Округленные средние значения рангов принимаются за итоговые оценки качества проектов. Проверку достоверности результата на определенном уровне значимости (обычно с вероятностью 95 или 99%) можно провести с использованием t -критерия Стьюдента или F -критерия Фишера [2].

Предлагаемое решение

Для практического использования предложенный подход может быть реализован в виде следующего алгоритма.

Шаг 1. Рассчитывается значение коэффициента конкордации по формуле (1) или с использованием соотношений (2, 3) при наличии связанных рангов. Если $W \geq 0,6$ то выполняется переход к шагу 7, иначе – к шагу 2.

Шаг 2. Принимается $s=0$ и $j=0$, где s - номер группы экспертов, оценки которых согласованы, j - индекс эксперта.

Шаг 3. Увеличивается значение $s = s+1$ и формируется множество M_s , описывающее новую группу экспертов. Выбирается эксперт с номером $j = j+1$ и полученный индекс j включается в множество M_s . Значения рангов j -го эксперта присваиваются рангам этой группы $r_i^{(s)} = r_{ji}$; $i = \overline{1, m}$. При $j=n$ выполняется переход к шагу 6, иначе - к шагу 4.

Шаг 4. Выбирается эксперт с индексом $j = j+1$. Для каждой группы экспертов, индексы которых входят в множество M_k , где $k = \overline{1, s}$, вычисляется значение группового коэффициента ранговой корреляции

$$\rho_k = 1 - \frac{6 \sum_{i=1}^m (r_i^{(k)} - r_{ji})^2}{m^3 - m}. \quad (4)$$

Шаг 5. Индекс j включается в каждое множество M_k , где $k = \overline{1, s}$, такое, что полученное значение $\rho_k \geq 0,6$. Для соответствующих групп экспертов вычисляются усредненные значения рангов объектов

$$r_i^{(k)} = \frac{\sum_{p \in M_k} r_{ip}}{|M_k|}, \quad (5)$$

где $|M_k|$ - мощность множества M_k , т.е. количество экспертов, входящих в k -ю группу. При $j=n$ выполняется переход к шагу 6, иначе - к шагу 4. Если таких групп нет (т.е. для всех $k = \overline{1, s}$ имеем $\rho_k < 0,6$), то изменяется индекс $j = j-1$ и производится переход к шагу 3.

Шаг 6. Выбирается группа t , в которую входит наибольшее число экспертов

$$m_t = \max |M_k|, \quad k = \overline{1, s}.$$

При наличии нескольких групп с одинаковым значением m_t выбирается такая, для которой значение итогового группового коэффициента ранговой корреляции ρ_t максимально. В качестве рангов для оценки объектов принимаются округленные значения $r_i^{(t)}$, $i = \overline{1, n}$.

Шаг 7. Производится оценка значимости вычисленного коэффициента на требуемом уровне достоверности. Используется критерий χ^2 для коэффициента конкордации или Стьюдента (Фишера) для группового коэффициента ранговой корреляции. Конец алгоритма.

Выводы

Предложенная методика позволяет проводить исследование качества программных проектов, выделив группы экспертов, оценки которых согласованы. Для практического использования разработан итерационный алгоритм формирования таких групп экспертов на основе вычисления необходимых статистических коэффициентов, что в конечном итоге позволяет автоматизировать процесс принятия решений.

Библиографический список

1. Кремер Н.Ш., Путко Б.А. Эконометрика. – М.: Юнити-Дана, 2010.
2. Айвазян С. А., Мхитарян В.С. Теория вероятностей и прикладная статистика. – М.:Юнити-Дана, 2006.
3. Вентцель Е.С., Овчаров Л.А. Теория вероятностей и ее инженерные приложения. – М.: Высшая школа, 2000.
4. Дубов А.М., Мхитарян В.С., Трошин Л.И. Многомерные статистические методы. – М.: Финансы и статистика, 2001.

УДК 004.72; ГРНТИ 50.39

ПРОГРАММНЫЙ МОДУЛЬ ДЛЯ ОРГАНИЗАЦИИ СЕТЕВОГО ВЗАИМОДЕЙСТВИЯ ПРОГРАММ И ПРОГРАММНЫХ СИСТЕМ В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ

Д.А. Перепелкин, М.А. Иванчикова, В.С. Бышов, И.Ю. Цыганов

Рязанский государственный радиотехнический университет,

Россия, Рязань, dmitryperpelkin@mail.ru

Аннотация. В работе разработан программный модуль, предназначенный для организации сетевого взаимодействия программной системы и визуальной среды динамического конфигурирования программно-конфигурируемых сетей с контроллером OpenFlow. Исследование эффективности и работоспособности предложенного подхода проведено в виртуальной среде Mininet.

Ключевые слова. Программно-конфигурируемые сети, программная система, визуальная среда, модуль сетевого взаимодействия, openflow.

SOFTWARE MODULE FOR ORGANIZATION OF NETWORK INTERACTION OF PROGRAMS AND SOFTWARE SYSTEMS IN SOFTWARE DEFINED NETWORKS

D.A. Perepelkin, M.A. Ivanchikova, V.S. Byshov, I.Yu. Tsyganov

Ryazan State Radio Engineering University,

Ryazan, Russia, dmitryperpelkin@mail.ru

Abstract. The software module, intended for organization of network interaction of the program system and the visual environment of dynamic configuring of software defined networks with OpenFlow controller, is developed in the work. The research of efficiency and operability of the offered approach is conducted in the virtual environment Mininet.

Keywords: software defined networks, program system, visual environment, network interaction module, openflow.

Программно-конфигурируемые сети (ПКС) – это управляемая и адаптируемая сетевая архитектура [1, 2]. ПКС идеально подходят для высоконагруженного динамического характера современных приложений. Данная архитектура разделяет функции управления сетью и функции пересылки пакетов. Для управления процессом обработки данных, передающихся по сети, применяется протокол OpenFlow. Протокол используется для управления сетевыми OpenFlow-коммутаторами с центрального устройства сети – контроллера. Этот вид управления дополняет работающую на OpenFlow-коммутаторе встроенную программу, осуществляющую построение маршрута, создание таблицы маршрутизации и другие функции. Маршрутизация – это процесс вычисления маршрута передачи пакетов между устройствами в сети. В ПКС функции маршрутизации выполняет контроллер. Маршрутизация реализуется на основе алгоритмов маршрутизации. OpenFlow-коммутаторы направляют трафик (потoki) в сети в соответствии с инструкциями контроллера. Вычисление маршрутов в ПКС основано на сборе и распространении служебной информации о состоянии сети. В результате обработки полученных данных о сетевой структуре контроллер строит модель текущего состояния сети в виде формального описания графа, вершины которого ставятся в соответствие узлам сети, ребра – линиям связи. Имея модель сети, алгоритмы маршрутизации вычисляют оптимальный маршрут передачи данных, используя показатели линий связи – метрики [3].

Задачей данной работы является разработка программного модуля для организации сетевого взаимодействия программной системы и визуальной среды динамического конфигурирования ПКС с контроллером OpenFlow. В качестве контроллера OpenFlow в работе использовался контроллер POX с открытым исходным кодом. Данный контроллер реализован на языке программирования Python 2.7. В качестве среды передачи данных использовалась виртуальная компьютерная сеть, функционирующая с помощью эмулятора Mininet [4].

Алгоритмы маршрутизации дополнительно были реализованы в виде программных модулей для контроллера POX.

На рисунке приведены основные классы разработанного программного модуля.

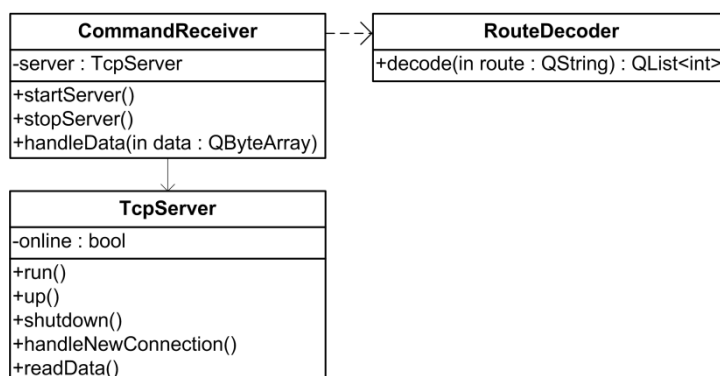


Рисунок – Основные классы для организации сетевого взаимодействия

Модуль сетевого взаимодействия решает задачу передачи данных из процесса контроллера POX в процесс программной системы и визуальной среды SDNTopology [5]. Класс **TcpServer** с помощью встроенных в библиотеку Qt типов создает и асинхронно с основным приложением прослушивает сетевой сокет. Управление экземпляром **TcpServer** осуществляется с помощью функций `run`, `up`, `shutdown`. Функция `handleNewConnection` регистрирует подключившегося клиента. В качестве клиента выступает процесс контроллера POX. Функция `readData` считывает данные, которые прислал клиент. Атрибут `online` определяет статус сервера.

Класс **CommandReceiver** находится в отношении ассоциации к классу **TcpServer**. Функции `startServer` и `stopServer` включают и выключают прослушивание сети. Функция `handleData` обрабатывает двоичные данные, полученные из процесса контроллера POX следующим образом. Полученные данные из двоичных, представленных встроенным типом библиотеки Qt **QByteArray**, преобразовывается в строку типа **QString**. Затем вызывается функция `decode` из класса **RouteDecoder**. Класс **RouteDecoder** предназначен для декодирования маршрута. Функция `decode` преобразует строку в список номеров вершин в маршруте. Этот список затем передается подсистеме рисования.

Использование данного программного модуля позволяет организовать эффективное и надежное сетевое взаимодействие программ и программных систем в ПКС на базе протокола OpenFlow, а также визуально и в режиме реального времени отобразить работу ПКС.

Работа выполнена при финансовой поддержке гранта РФФИ 16-47-620300 p_a.

Библиографический список

1. N. McKeown, T. Anderson, H. Balakrishnan et al., "Openflow: enabling innovation in campus networks", ACM SIGCOMM Computer Communication Review, 2008, vol. 38, no. 2, pp. 69-74.
2. Hilmi Enes Egilmez, Adaptive Video Streaming over OpenFlow Networks with Quality of Service, Thesis for Degree of Master Science in Electrical and Electronics Engineering, Koc University, 2012, 91 p.
3. Koryachko, V., Perepelkin, D., Ivanchikova, M., Byshov, V., and Tsyganov, I., Analysis of QoS metrics in software defined networks, Proc. of the 6-th Mediterranean Conference on Embedded Computing (MECO), 2017, pp. 374–378.
4. "Virtual Network Mininet", <http://mininet.org>.
5. Perepelkin D., Byshov V., "Visual Design Environment of Dynamic Load Balancing in Software Defined Networks". Proceedings Radioelektronika 2017 – IEEE 27th International Conference Radioelektronika 2017 (Radioelektronika-2017), 2017, pp. 1-4. DOI: 10.1109/RADIOELEK.2017.7936643.

УДК 004.72; ГРНТИ 50.39

СРАВНЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ, РЕАЛИЗУЮЩЕГО АЛГОРИТМ КЛАСТЕРИЗАЦИИ K-MEANS И ЕГО МОДИФИКАЦИИ

Л.С. Дашкина

*Рязанский государственный радиотехнический университет,
РФ, Рязань, lily.dashkina95@mail.ru*

Аннотация. В данной работе рассматривается алгоритм кластеризации k-means и его модификации, программное обеспечение реализующее их и проводится сравнение этих алгоритмов на различных данных.

Ключевые слова. Интеллектуальный анализ данных, кластеризация, k-means

COMPARISON OF THE SOFTWARE IMPLEMENTING THE K-MEANS CLUSTERING ALGORITHM AND ITS MODIFICATION

L. S. Dashkina

*Ryazan State Radioengineering University,
Russia, Ryazan, lily.dashkina95@mail.ru*

Abstract. This paper deals with k-means clustering algorithm and its modifications, the software that implements them and compares these algorithms on various datasets.

Keywords. Data mining, cluster analysis, k-means

Кластерный анализ позволяет рассматривать достаточно большой объем информации и резко сокращать, сжимать большие массивы информации, делать их компактными и наглядными. Его используют, когда отсутствуют априорные сведения относительно классов, к которым можно отнести объекты исследуемого набора данных, либо, когда число объектов велико, что затрудняет их ручной анализ. Так, кластеризация применяется в информационном поиске, розничной торговле, телекоммуникациях, страховании, сегментации изображений [3, 4, 5].

На сегодня предложено несколько десятков алгоритмов кластеризации и еще больше их разновидностей. Несмотря на это, в Data Mining в первую очередь применяются алгоритмы, которые понятны и просты в использовании. Один из них это алгоритм k-means. Алгоритм состоит из 4 шагов [4]:

1) задается число кластеров k , которое должно быть сформировано из объектов исходной выборки;

2) выбирается k записей, которые будут служить начальными центрами кластеров. Начальные точки, из которых потом «вырастет» кластер, часто называют «семенами». Каждая такая запись представляется собой своего рода «эмбрион» кластера, состоящий из одного элемента;

3) для каждой записи исходной выборки определяется ближайший к ней центр кластера. Правило, по которому вычисляется расстояние в многомерном пространстве признаков, называется метрикой;

4) производится вычисление центроидов – центров тяжести кластеров. Это делается путем определения среднего для значений каждого признака всех записей в кластере. Затем старый центр кластера смещается в его центроид. Таким образом, центроиды становятся новыми центрами кластеров для следующей итерации алгоритма.

Шаги 3 и 4 повторяются до тех пор, пока выполнение алгоритма не будет прервано либо пока не будет выполнено условие в соответствии с некоторым условием сходимости.

Достоинства алгоритма k-means:

1) алгоритм имеет высокую скорость работы в случае выбора оптимальных начальных значений центров масс кластеров. Вычислительная сложность алгоритма k -

means, как и у других неиерархических алгоритмов оценивается как $O(nkl)$, где k – количество кластеров, l – число итераций;

2) устойчивость: алгоритм является устойчивым к погрешностям во входных данных, так как при вычислении центров кластеров расстояния между объектами усредняются, что приводит к уменьшению ошибки. Алгоритм также является устойчивым к погрешностям, допускаемых при вычислениях, так как операция сложения на каждой новой итерации выполняется над входными данными.

Недостатки алгоритма k-means:

1) число кластеров является входным аргументом алгоритма. Таким образом, некорректный выбор данного параметра может привести к плохим результатам работы алгоритма, поэтому зачастую проводят несколько прогонов алгоритма с разными значениями k , чтобы подобрать оптимальный;

2) алгоритм слишком чувствителен к выбросам, которые могут исказить среднее. Возможным решением этой проблемы является использование модификации алгоритма - алгоритм k-means++;

3) алгоритм может медленно работать на больших базах данных. Возможным решением данной проблемы является использование выборки данных.

Одна из модификаций алгоритма — k-means++. Суть улучшения заключается в нахождении более «хороших» начальных значений центроидов кластеров. Оригинальный k-means не регламентирует то, как выполняется этот этап алгоритма, и поэтому является нестабильным.

Вместо того, чтобы принимать среднее значение (mean) объекта в кластере в качестве контрольной точки может использоваться медиана (medoid), которая является наиболее централизованным объектом в кластере. Наиболее распространенной реализацией k-медоидной кластеризации является алгоритм Partitioning Around Medoids (PAM). Ограничение использования алгоритма это небольшой объем данных. Достоинства этого алгоритма те же, что и у k-means, но только PAM менее чувствителен к выбросам в сравнении.

На сегодняшний день существует множество программного обеспечения, которое реализует алгоритмы кластеризации: проприетарное (Matlab, RapidMiner, SPSS и др.) и открытое. В данной статье будут рассмотрены открытые приложения Deductor, Weka и ELKI.

BaseGroup Labs [7] — поставщик программных продуктов и решений в области анализа данных, специализирующийся на разработке систем для глубокого анализа данных, охватывающих вопросы сбора, интеграции, очистки данных, построения моделей и визуализации. Deductor является платформой, на базе которой создаются законченные аналитические решения. Платформа ориентирована на применение экспертами в различных предметных областях, позволяет обрабатывать любую структурированную табличную информацию. Deductor Academic Studio доступен для открытого скачивания, работает только под операционной системой Windows. В Deductor Academic 5.3 реализован алгоритм k-means со случайным выбором первых центроидов кластеров и одна метрика - евклидова. После проведения кластеризации доступны:

- профили кластеров, включающие основные статистические характеристики по каждому атрибуту (среднее, стандартное отклонение, стандартная ошибка...);
- таблица, содержащая исходные данные и два новых столбца: номер кластера, к которому относится экземпляр, и расстояние до центра кластера;
- диаграмма размещения, на которой отображены экземпляры данных на плоскости (2 атрибута) или в пространстве (3 атрибута) и др.;

ELKI [8] - программное обеспечение для интеллектуального анализа данных с открытым исходным кодом (AGPLv3), написанное на Java. В центре внимания ELKI лежит исследование алгоритмов кластеризации. ELKI спроектирован так, что его можно легко расширить для исследователей и студентов в этой области и приветствует вклад дополнительных

методов. ELKI стремится предоставить большой набор алгоритмов. В ELKI 0.7.1 реализовано целое множество модификаций k-means (k-meansSort, k-medoidsPAM, x-means и др.) с 9 вариантами выбора начальных центроид, включая k-means++, и свыше 20 метрик. Кроме того в ELKI реализована хорошая визуализация результатов кластеризации (диаграммы размещения и диаграмма Вороного) и оценки качества кластеризации (индекс Жаккарда, оценка F1, ...). ELKI использует SVG формат для визуализации, имеющий преимущества для экспорта печати, и Apache Batik для рендеринга.

Weka [6] был разработан в Университете Вайкато в Новой Зеландии. Система написана на Java и распространяется в соответствии с условиями GNU GPL. Она обеспечивает единый интерфейс для многих различных алгоритмов обучения, наряду с методами предварительной и постобработки и для оценки результата схемы обучения по любому данному набору данных. В Weka 3.8 реализован алгоритм k-means как со случайным выбором начальных центроид, так и по алгоритму k-means++ и доступно 5 метрик. Визуализатор Weka Explorer включает диаграмму размещения и основные статистические данные о кластерах.

В Weka в качестве набора данных помимо CSV-файлов, используется специальный формат – ARFF. Файл ARFF (Attribute-Relation File Format) - текстовый файл ASCII, который описывает список экземпляров, совместно использующих набор атрибутов. Файлы ARFF имеют два отдельных раздела: заголовок и данные. Заголовок содержит название набора данных (relation), список атрибутов (attribute) с указанием их типов (numeric, string, date...). В разделе данных (data) хранятся значения атрибутов для каждого экземпляра, разделенные запятыми, как и в CSV-файлах [8].

Для сравнения алгоритмов использовалось три набора данных (два последних взяты из Weka):

1. Список точек на плоскости с указанием координат x и y. Точки визуально легко разделить на 4 кластера. Этот набор данных использовался для начальной проверки;

2. Набор данных об ирисах, с которыми в середине 30-х гг. прошлого столетия работал статистик Р. А. Фишер (ирисы Фишера). Он рассмотрел три класса ирисов: *Iris setosa*, *Iris versicolor* и *Iris virginica*. Для каждого из них было представлено по 50 экземпляров с разными значениями четырех параметров: длина и ширина чашелистника, длина и ширина лепестка [1];

3. Набор данных о производительности процессоров, включающий следующие атрибуты [8]:

- время машинного цикла в наносекундах (machine cycle time in nanoseconds);
- минимальная основная память в килобайтах (minimum main memory in kilobytes);
- максимальная основная память в килобайтах (maximum main memory in kilobytes);
- кэш-память в килобайтах (cache memory in kilobytes);
- минимальное число каналов памяти (minimum channels in units);
- максимальное число каналов памяти (maximum channels in units).

В качестве оценки результатов кластеризации помимо реализованных в самих приложениях характеристик использовались диаграммы средних и оценки дисперсий [4]. На основе этих оценок и вышеуказанных возможностей приложений можно сделать вывод, что лучшей программой для проведения кластерного анализа с использованием алгоритма k-means и его модификаций из рассмотренных является ELKI, так как в ней реализовано большее количество модификаций алгоритма и предусмотрена более наглядная визуализация результатов кластеризации.

Библиографический список

1. Барсегян А. А., Куприянов М. С., Холод И. И., Тесс М. Д., Елизаров С. И. Анализ данных и процессов: учеб. пособие. 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 512 с.
2. Маннинг К. Д., Рагхаван П., Шютце Х. Введение в информационный поиск.: Пер. с англ. — М.: ООО «И.Д. Вильямс», 2011. — 528 с.
3. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям: учебное пособие. 2-е изд., испр. — СПб.: Питер, 2013. — 704 с.
4. Ту Дж., Гонсалес Р. Принципы распознавания образов. Пер. с англ. И. Б. Гуревича под ред. Ю. И. Журавлева. М.: Мир, 1978. — 414 с.
5. Machine Learning Group at the University of Waikato [Электронный ресурс], URL: <https://www.cs.waikato.ac.nz/ml/index.html> (дата обращения 10.02.18)
6. Deductor Продвинутой аналитики без программирования [Электронный ресурс], URL: <https://basegroup.ru/deductor/description> (дата обращения 10.02.18)
7. ELKI: Environment for Developing KDD-Applications [Электронный ресурс], URL: <https://elki-project.github.io/> (дата обращения 10.02.18)

УДК 004.021; ГРНТИ 50.53.57

АЛГОРИТМ АВТОМАТИЧЕСКОГО ОПРЕДЕЛЕНИЯ КОЛИЧЕСТВА КЛАСТЕРОВ В ЗАДАЧЕ СЕГМЕНТАЦИИ ГИПЕРСПЕКТРАЛЬНЫХ СНИМКОВ

М.Н. Гурьева

АО «Сбербанк-технологии»

Россия, Рязань, maria_guryeva@mail.ru

Аннотация. В работе рассмотрена задача кластеризации гиперспектрального снимка. Разработан алгоритм автоматического определения количества кластеров с использованием чисел Фибоначчи. Исследовано влияние первоначальной центральной выборки на качество кластеризации. Приведена адаптация метода нейронной карты Кохонена на трехмерные данные.

Ключевые слова. Кластеризация, числа Фибоначчи, нейронная карта Кохонена, гиперспектральные изображения.

ALGORITHM OF AUTOMATIC DETERMINATION OF QUANTITY CLUSTERS IN THE PROBLEM OF SEGMENTATION OF HYPERSPECTRAL SNAPSHOTS

M.N. Guryeva,

Sberbank-Technologies

Russia, Ryazan, maria_guryeva@mail.ru

Abstract. The task of clustering a hyperspectral image is considered. An algorithm for automating the determination of the number of clusters using Fibonacci numbers is developed. The influence of the initial central sample on the quality of clustering was investigated. The adaptation of the Kohonen neural map method to three-dimensional data is presented.

Keywords. Clustering, Fibonacci numbers, Kohonen neural map, hyperspectral images.

Введение

Гиперспектральные изображения (ГСИ) представляют собой набор данных о регистрируемой на момент съемки спектральной плотности энергетической яркости (СПЭЯ) поверхности Земли на входном зрачке аппаратуры, содержащий в себе изображения одной и той же сцены в большом количестве узких соприкасающихся спектральных диапазонов. В результате дополнения каждой пространственной координаты снимка её спектральной характеристикой (СХ), формируется трехмерный массив данных, описывающий распределение энергии излучения по длине волны каждой отображаемой точки регистрируемой местности. Таким образом, каждая точка на изображении трансформируется в вектор данных, количест-

во отсчетов которого возрастает с увеличением спектрального разрешения оптоэлектронной системы регистрации ГСИ. Спектральное разрешение ГСИ на порядок выше, чем у мультиспектральных изображений, что позволяет с большей точностью построить СХ анализируемой области (рис. 1) [1].

Кластеризация таких данных позволяет идентифицировать на изображении объекты различной природы, например, растительность, водная поверхность, открытая почва, антропогенная зона. При детальном анализе каждого кластера можно выделить зоны неоднородности спектральной характеристики, определяющие изменение количественных и качественных характеристик объекта. Например, в [2] представлен алгоритм идентификации загрязнений приповерхностного водного слоя по данным видеонаблюдения на основе оценки концентрации минеральных и органических веществ.

Наряду с классическими методами кластеризации, такими как k-means, ISODATA в настоящее время для широко применяются методы искусственного интеллекта, например, нейронная карта Кохонена. Большинство алгоритмов кластеризации являются автоматизированными и требуют предварительного задания оператором количества сегментов, что неоптимально, поскольку подвержено влиянию человеческого фактора и исключает возможность применения в системах реального времени. В настоящей работе представлен алгоритм автоматического определения количества кластеров на изображении как модернизация ранее разработанной адаптации нейронной карты Кохонена на трехмерные данные для кластеризации ГСИ.

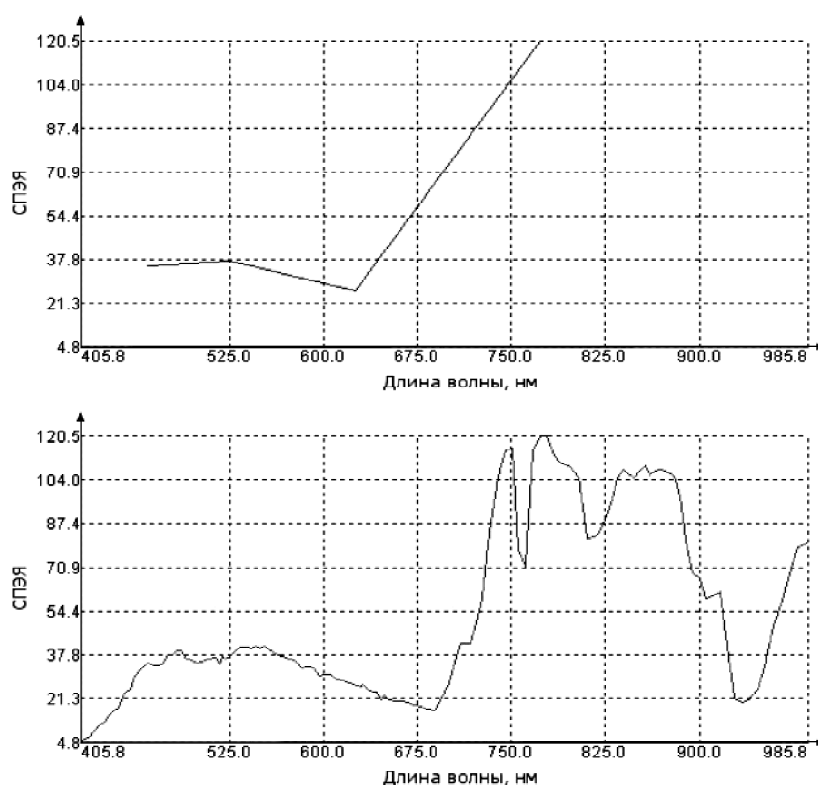


Рис. 1. Спектральная характеристика на мультиспектральных и гиперспектральных данных

Нейронная карта Кохонена

Карта представляет собой нейронную сеть с обучением без учителя, содержащую один слой. Итогом работы карты является проецирование многомерного пространства на пространство с более низкой размерностью, в данном случае, двумерное. Преимуществами

этого метода являются устойчивость к зашумленным данным, быстрое и неуправляемое обучение, возможность упрощения многомерных входных данных с помощью визуализации [3]. В процессе анализа сетью большой входной выборки происходит вычисление общих признаков и группировка данных на кластеры.

Первым этапом является формирование первичной выборки. В классической реализации карты Кохонена первичная выборка представлена последовательностью векторов, заполненная случайными числами.

Вторым этапом является настройка карты, результатом которого является формирование центральной выборки для дальнейшей кластеризации ГСИ. Последовательность СХ пикселей подается на вход корректировочного алгоритма. Для каждой следующей анализируемой точки определяется расстояние текущей спектральной характеристики до всех векторов первичной выборки. Вектор, расстояние до которого минимально, признается центральной корректировочной точкой и происходит подстройка карты:

$$m(t) = m(t-1) + h_c(t) \cdot (x(t) - m(t-1))$$

$$h_c(t) = \alpha(t) \cdot \delta(t)$$

$$\delta(t) = e^{-\frac{R}{2\sigma^2(t)}},$$

где $m(t)$ – скорректированный вектор;
 $m(t-1)$ – нескорректированный вектор;

$x(t)$ – СХ текущего пикселя;

$h_c(t)$ – корректировочная функция;

$\alpha(t)$ – скорость обучения;

$\delta(t)$ – функция соседства, определяющая количество соседних центроидов, которые затрагивает подстройка карты;

R – расстояние на карте между центральной корректировочной точкой и корректируемым вектором.

Скорость обучения определяет точность подстройки и на начальном этапе принимает равной 0,7. В процессе обучения сети этот коэффициент падает по экспоненциальной зависимости. Это означает, что на первых итерациях происходит жесткая корректировка, а на последних – «подстройка», что в конечном итоге приводит к формированию усредненной СХ каждого класса [4].

На третьем этапе происходит оценка принадлежности каждого пикселя ГСИ к тому или иному классу по критерию минимального расстояния до центроида. Таким образом, метод является классификатором ГСИ на группы пикселей с СХ близких признаков, в результате работы двумерная карта, разбитая на ячейки в соответствие с принадлежностью пикселей к тому или иному классу.

Алгоритм автоматического определения количества кластеров

В контексте ГСИ формирование первичной выборки в виде последовательности случайных векторов является неоптимальным, поскольку СХ наблюдаемых объектов имеют определенную известную форму, например, водная поверхность имеет слабую отражательную способность в инфракрасной зоне, антропогенные области, напротив, высокую, а область лесополосы имеет два максимума СХ – в зеленой и инфракрасной зоне спектра.

В настоящей реализации входная последовательность выбирается как набор СХ пикселей, распределенных по изображению по методу чисел Фибоначчи, адаптированного на двумерный случай. Это представляет собой метод равномерного распределения точек, за-

ключающийся в итеративном делении изображения на n частей по длине и ширине и взятию СХ центральной точки каждой части для анализа.

$$n_0 = 1, n_1 = 2, \dots, n_i = n_{i-1} + n_{i-2}.$$

Целью такого итеративного накопления является формирование выборки СХ, максимально удаленных друг от друга по Евклидову расстоянию. На каждой итерации центральная выборка дополняется по следующему алгоритму. Для каждого анализируемого вектора последовательности рассчитывается Евклидово расстояние до всех векторов центральной выборки, весовым коэффициентом данного вектора принимается минимальное из них:

$$W = \min \{E_1, E_2, \dots, E_k\}$$

где k – количество векторов текущей центральной выборки.

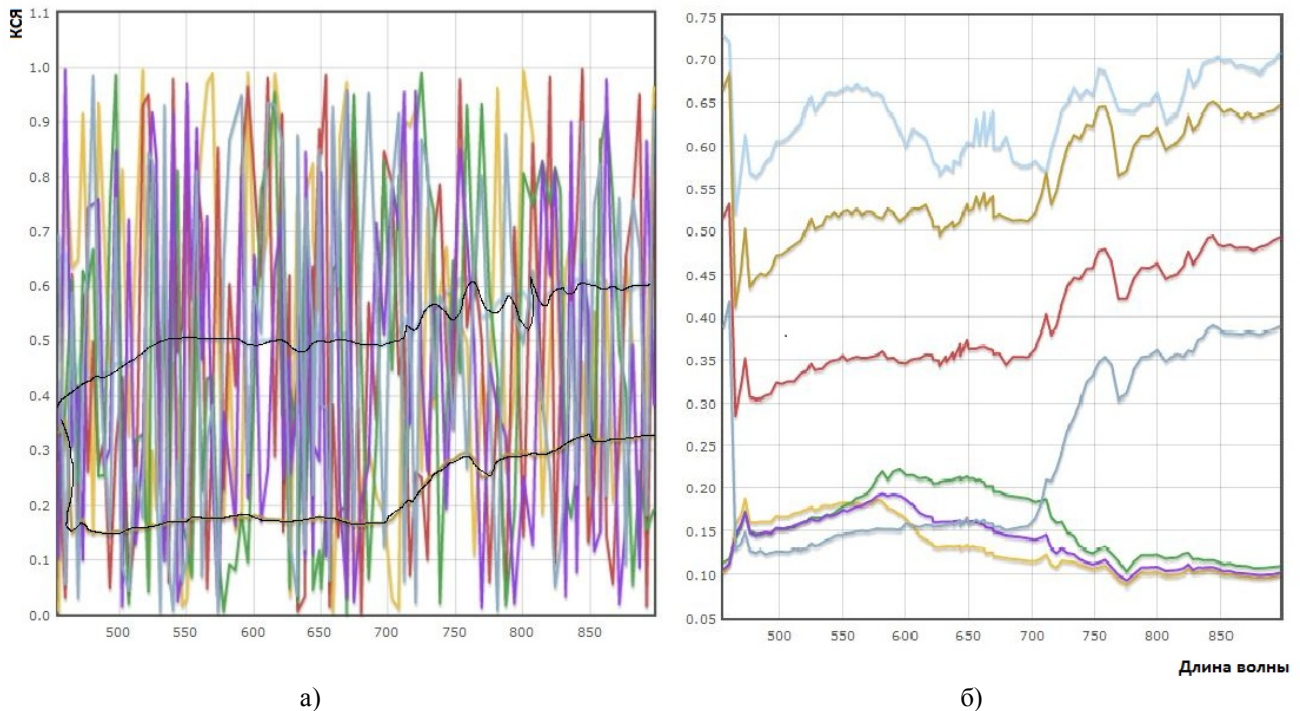
По окончании итерации имеется n весовых коэффициентов. СХ, соответствующие максимальным значениям $W_i > T$, где T – некоторое пороговое значение, добавляются к центральной выборке, после чего итерация повторяется для n_{i+1} входных параметров.

Критерием окончания поиска является резкое падение этого показателя по отношению к предыдущей итерации. Таким образом, на ГСИ автоматически определяется k характеристик, каждая из которых на следующем этапе является начальным центроидом для кластеризации.

Экспериментальные исследования

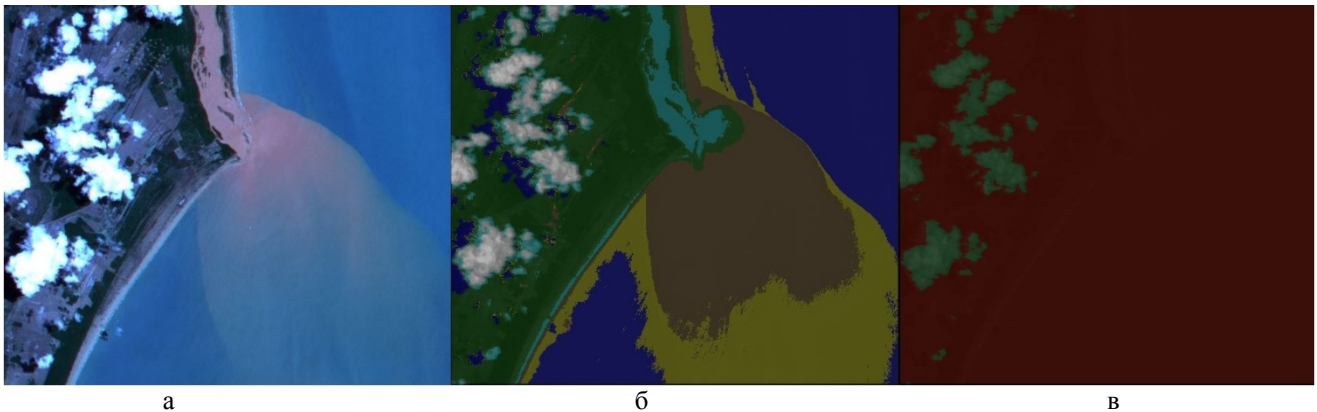
Для доказательства эффективности вышеописанного алгоритма был проведен сравнительный анализ кластеризации ГСИ нейронной картой Кохонена с первичной выборкой двух видов: СХ, заполненные случайными числами соответственно разрядности изображения; реальные СХ, определенные алгоритмом с автоматическим поиском количества кластеров. При этом количество классов для выборки случайных векторов было задано соответственно автоматически определенному. Изображение предварительно прошло радиометрическую и атмосферную коррекцию, а также фильтрацию Винера для устранения шума.

Форма СХ центральной выборки после настройки карты для обоих случаев приведена на рисунке 2. Видно, что при формировании первичной выборки предлагаемым алгоритмом центроиды для дальнейшей кластеризации формируются более точно, в отличие от варианта первичной выборки, заполненной случайными числами. Также можно отметить, при формировании первичной выборки случайными векторами из семи искомым реальную форму СХ после настройки карты приняли только два, что доказывает важность первичной выборки для кластеризации описанным методом.



а) б)
Рис. 2. Центральная выборка после настройки карты:
а) при задании первичных СХ в виде случайных векторов,
б) при автоматическом определении первичных СХ

Результаты кластеризации ГСИ приведены на рисунке 3. Кластеризацией по центральной выборке, определенной вышеописанным алгоритмом в автоматическом режиме было распознано 3 области водного объекта, суша и облачность, в то время как случайной выборкой после настройки успешно была идентифицирована только облачность.



а б в
Рис. 3 – Результаты кластеризации ГСИ:
а) RGB-модель исходных данных;
б) при задании первичных СХ в виде случайных векторов;
в) при автоматическом определении первичных СХ

Таким образом, можно отметить заметное влияние формы первичной выборки для дальнейшей настройки центроидов на качество кластеризации

Численные результаты эксперимента приведены в таблице 1. Рассчитано среднее внутрикластерное расстояние СХ до центроидов и межкластерная дисперсия. Можно отметить значительное падение среднего внутрикластерного расстояния и межкластерной дисперсии при применении алгоритма автоматического поиска первичной выборки.

Таблица 1. Численные результаты эксперимента

Тип первичной выборки	Среднее внутрикластерное расстояние		Средняя межкластерная дисперсия
Автоматическое определение	Кластер 1	0.35	0,16
	Кластер 2	0.71	
	Кластер 3	0.67	
	Кластер 4	0.27	
	Кластер 5	0.08	
	Кластер 6	0.38	
	Кластер 7	0.42	
Случайные векторы	Кластер 1	1,03	0,68
	Кластер 2	0,86	
	Кластер 3	Не обнаружен	
	Кластер 4	Не обнаружен	
	Кластер 5	Не обнаружен	
	Кластер 6	Не обнаружен	
	Кластер 7	Не обнаружен	

Выводы

Применение разработанного алгоритма вычисления центральной выборки позволило в 3.5 раза увеличить количество обнаруженных классов объектов, уменьшить среднее внутрикластерное расстояние и межкластерную дисперсию в 1.5 и 4.25 раза соответственно, что говорит о его эффективности. Кроме того, разработанный алгоритм автоматически оценивает оптимальное количество распознаваемых классов объектов, что может позволить использовать изложенный подход в системах реального времени без участия оператора.

Дальнейшее увеличение количества распознанных кластеров возможно повышением чувствительности алгоритма за счет предварительной обработки изображения с целью избавления от шума, модернизации вычисляемой меры близости векторов, снижения порога оптимальности количества кластеров.

Разработанные метод и алгоритмы могут быть применены в «обратных» задачах распознавания составляющих поверхностей космических аппаратов и природных объектов со станций на поверхности Земли [5].

Библиографический список

1. Демидова Л. А., Еремеев В. В., Мятлов Г. Н., Тишкин Р. В., Юдаков А. А. Сегментация объектов по данным гиперспектральной съёмки Земли с использованием методов искусственного интеллекта, Вестник Рязанского государственного радиотехнического университета. 2014. №3(49). С. 3-10.
2. Григорьева О. В., Шилин Б. В. Опыт оценки экологических характеристик акваторий морских портов по данным видеоспектральной съемки, Современные проблемы дистанционного зондирования Земли из космоса. 2012. Т. 9. № 1. С. 156–166
3. Горбаченко В. И. Сети и карты Кохонена: [Электронный ресурс], URL: Научно-исследовательский центр самоорганизации и развития систем. М., 2010-2014. URL: <http://gorbachenko.self-organization.ru/>
4. Гурьева М. Н. Применение самоорганизующейся карты Кохонена для сегментации гиперспектральных изображений, Юбилейная 25-я международная конференция "GRAPHICON 2015" АНО научного общества "ГРАФИКОН" Института физико-технической информатики, Протвино, 22-25 сентября 2015 г.: Труды. 2015. С. 93-95.
5. Миронов В.В., Муртазов А.К., Усовик И.В. Системные методы мониторинга околоземного космического пространства /под науч. ред. проф. Миронова В.В. - Рязань: Book Jet, 2017. 318 с. (в формате А4).

УДК 004.89; ГРНТИ 27.41.41

ОСОБЕННОСТИ ГРАФИЧЕСКОГО ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ПРОГРАММЫ БАЛАНСИРОВКИ НАГРУЗКИ В СЕТЯХ ЦЕНТРОВ ОБРАБОТКИ ДАННЫХ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

А. Сапрыкин, Е. Сапрыкина

*Рязанский государственный радиотехнический университет,
Россия, Рязань, saprykin.a.n@cnit.rsreu.ru*

Аннотация. В данной работе рассмотрены особенности графического пользовательского интерфейса программы балансировки нагрузки в сетях центров обработки данных на основе генетического алгоритма. Описаны возможности разработанной программы, ее основные вкладки и разделы. Приведены графические результаты оптимизации нагрузки сети.

Ключевые слова. Сети ЦОД, разработка пользовательского интерфейса, программа балансировки нагрузки, генетический алгоритм.

FEATURES OF THE GRAPHICAL USER INTERFACE OF THE LOAD BALANCING PROGRAM IN NETWORKS OF DATA CENTERS BASED ON A GENETIC ALGORITHM

A. Saprykin, E. Saprykina

*Ryazan State Radio Engineering University,
Russia, Ryazan, saprykin.a.n@cnit.rsreu.ru*

Abstract. This paper explores the features of the graphical user interface of the load balancing software in the networks of data centers based on the genetic algorithm. The possibilities of the developed program, its main tabs and sections are described. Graphical results of network load optimization are given.

Keywords. Datacenter networks, user interface development, load balancing software, genetic algorithm.

На данный момент большинство центров обработки данных характеризуются ячеистой сетевой архитектурой, топология которой близка к полносвязной. Работы [1,2] рассматривают возможности использования двухфазной маршрутизации, которая подходит для интегрирования с данной сетевой топологией [3].

Центры обработки данных выстраивают свою сетевую топологию таким образом, чтобы обеспечить высокую производительность при наличии группы взаимосвязанных серверов. При этом между каждыми двумя отдельными серверами сетевое соединение поддерживается одновременно несколькими сетевыми каналами, которые объединяют всю совокупность вычислительных ресурсов. В таком случае суммарный объем внутреннего сетевого трафика превосходит объем внешнего в несколько раз [4].

Центры обработки данных поддерживают различные модели сетевого трафика: “потoki-мыши” и “потoki-слоны”. Обе данные модели характеризуются различным поведением и индивидуальными требованиями, что может приводить к возникновению задержек в сети и даже потере пакетов.

Для балансировки различных по требованиям сетевых потоков в центрах обработки данных авторами была разработана программа балансировки нагрузки с использованием генетического алгоритма. Программное средство было разработано на языке Object Pascal с помощью открытой среды разработки программного обеспечения Lazarus, вследствие чего оно является кроссплатформенным и может быть использовано в таких операционных системах, как Linux, Windows или Mac OS.

В данной статье рассматриваются особенности графического пользовательского интерфейса разработанной программы.

Интерфейс программы состоит из 6 основных вкладок:

1. Главная вкладка (рис. 1). Содержит такие разделы, как параметры вывода промежуточной информации, начальные параметры алгоритма балансировки, меню выбора стратегии

отбора лучших особей алгоритма и окно вывода наилучших результатов. Также данная вкладка отображает полосы прогресса процедуры формирования начальной популяции алгоритма и процедуры поиска наилучшего решения.

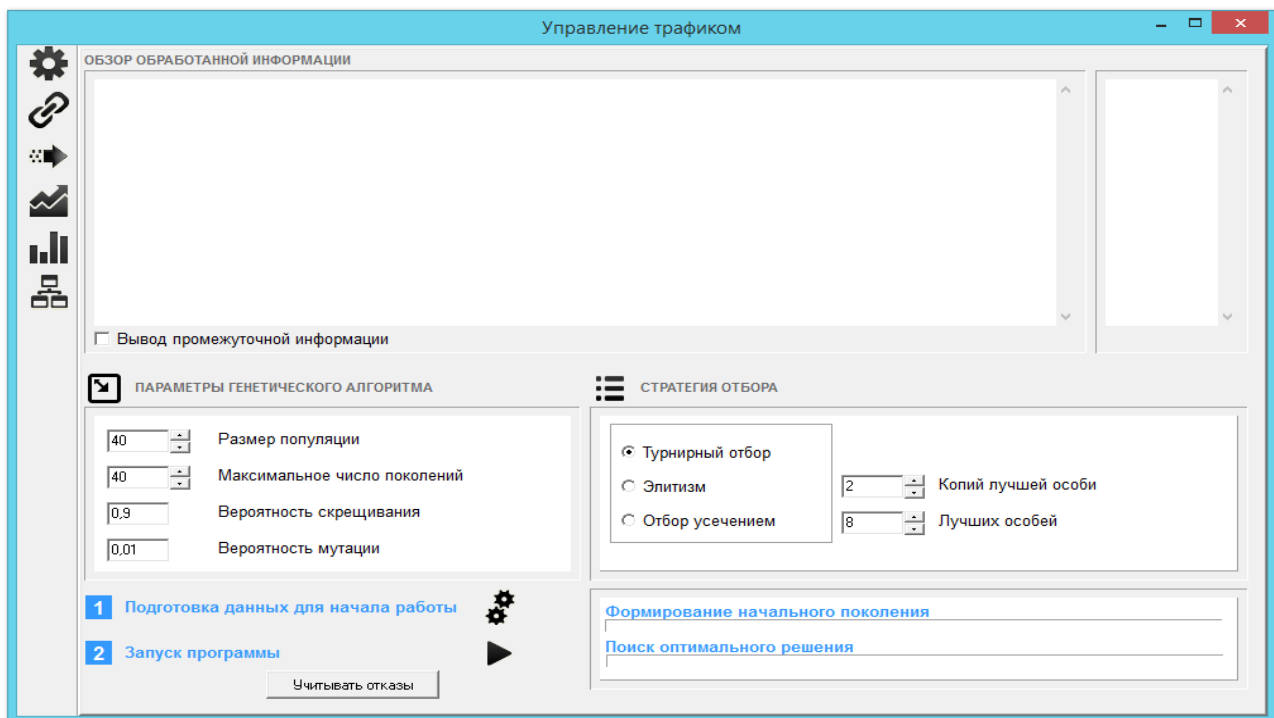


Рис. 1. Главная вкладка программы балансировки нагрузки в сетях центров обработки данных

2. Вкладка настройки сетевых связей.
3. Вкладка настройки сетевых потоков.
4. Вкладка графического отображения результатов работы алгоритма, содержащая значения фитнес-функции для каждой итерации. Пример графика фитнес-функции представлен на рисунке 2.

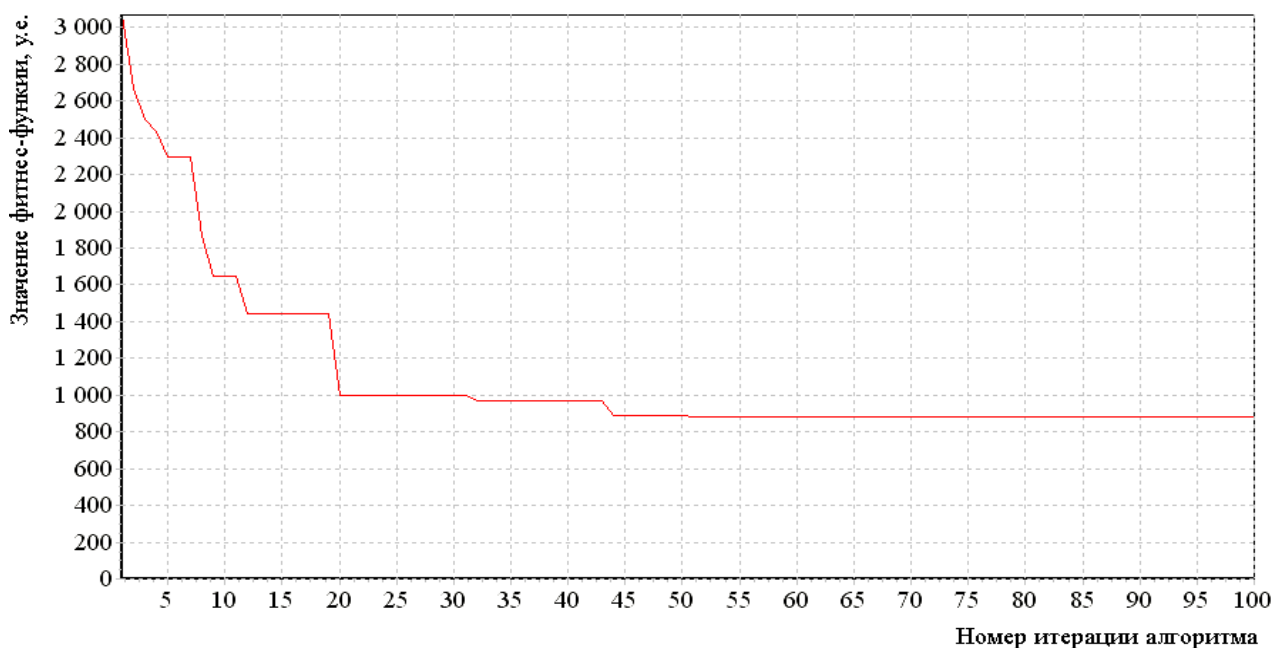


Рис. 2. График фитнес-функции генетического алгоритма

5. Вкладка графического отображения нагрузки на сетевые каналы до и после оптимизации. Пример графиков нагрузки сетевых каналов представлен на рисунке 3. Состояние загрузки каналов до оптимизации представлено в левой части рисунка и обозначено желтым цветом, после оптимизации – в правой части рисунка и обозначено зеленым цветом.

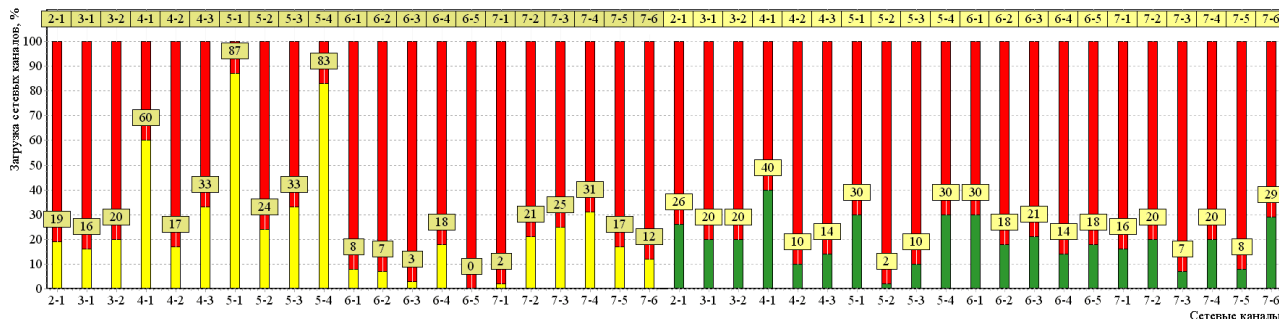


Рис. 3. График загрузки сетевых каналов до и после оптимизации

5. Вкладка визуализации построения сетевой топологии. Пример визуализированной схемы сети представлен на рис. 4. Синим и зеленым цветами обозначены узлы двух разных подсетей.

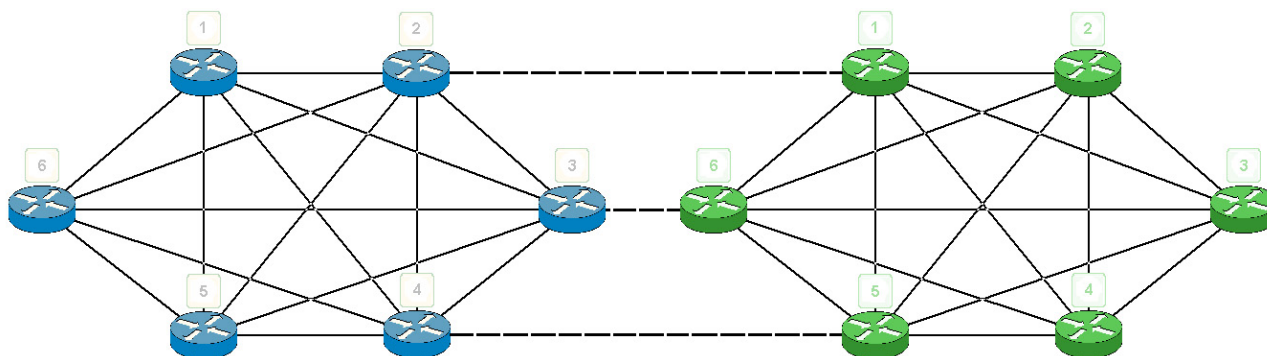


Рис. 4. Визуализированная схема сети

Для оптимизации нагрузки в сетях центров обработки данных авторами был использован генетический алгоритм. Для повышения эффективности оптимизации программа позволяет производить настройку основных параметров генетического алгоритма, таких как количество особей в каждой популяции, максимальное число итераций алгоритма, вероятность мутации особей, вероятность скрещивания особей, метод отбора особей в новую популяцию, максимальная длина хромосомы.

В ходе тестирования программы было проведено несколько экспериментов, которые подтвердили корректность работы разработанного алгоритма в соответствии с поставленными задачами и его эффективность. В качестве тестовых моделей были использованы полносвязные сетевые топологии с количеством узлов до 32 и количеством сетевых потоков до 50. Оптимизация, осуществленная программой, существенно снизила нагрузку наиболее занятых сетевых каналов, равномерно распределив ее между всеми каналами сети. Таким образом происходит снижение сетевых задержек и появляется возможность создания резервной полосы пропускания.

Библиографический список

1. Ижванов Ю.Л., Корячко В.П., Шибанов А.П., Сапрыкин А.Н., Лукьянов О.В. Оптимизация сети с дозированной балансировкой нагрузки // Системы управления и информационные технологии № 3(49). 2012. С. 37-42.
2. Корячко В.П., Шибанов А.П., Сапрыкин А.Н. Расчет вариации времени передачи пакетов в сети с двухфазной маршрутизацией: Труды XII всероссийского совещания по проблемам управления. Электрон, дан. -М.: ИПУ РАН, 2014.
3. Сапрыкина, Е.Н. Использование двухфазной многоканальной маршрутизации в сетях дата-центров / Е.Н. Сапрыкина, А.Н. Сапрыкин // Новые информационные технологии в научных исследованиях: материалы XXI Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. – Рязань: РГРТУ, 2016. – С. 337–338.
4. Greenberg A., Hamilton J. R., Jain N., Kandula S., Kim C., Lahiri P., Maltz D.A., Patel P., and Sengupta S., "VL2: A scalable and flexible data center network," vol. 54, no. 3, pp. 95–104, Mar. 2011.

УДК 004.891.2; ГРНТИ 28.23.35

СВОБОДНО РАСПРОСТРАНЯЕМЫЕ ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ СОЗДАНИЯ ЭКСПЕРТНЫХ СИСТЕМ

В.П. Вишневская

*Рязанский государственный радиотехнический университет
veronikapvishnevskaya@gmail.com*

Аннотация. В работе представлен обзор программных средств для создания экспертных систем. Определяются требования и критерии к разрабатываемой системе.

Ключевые слова. Экспертные системы, базы знаний, базы данных, состав экспертных систем.

FREE PROGRAM SOFTWARE FOR CREATING EXPERT SYSTEMS

V.P. Vishnevskaya

*Ryazan State Radio Engineering University
veronikapvishnevskaya@gmail.com*

Annotation. The paper provides an overview of software tools for creating expert systems. The requirements and criteria for the system developed are determined.

Keywords. Expert systems, knowledge bases, databases, composition of expert systems.

При поступлении в вуз у абитуриентов возникает множество вопросов, таких как, какие факультеты есть в вузе, какие предметы необходимо сдавать для поступления, перечень требуемых документов и многие другие вопросы, которые возникают в процессе поиска ответа на предыдущие.

В связи с данной проблемой возникает вопрос, что можно использовать для получения более конкретной и своевременной информации по интересующим вопросам.

Использование экспертных систем для получения ответов на интересующие вопросы может улучшить результат получения информации по сравнению с разделами «F.A.Q.». Данные системы используют экспертные знания для эффективного решения неформализованных задач, предоставляя пользователю самую актуальную информацию на момент выполнения запроса.

Экспертной системой называют программное обеспечение, позволяющее использовать полученные знания для решения неформализованных задач в узкой специальности [1].

Экспертная система состоит из решателя, базы данных(БД), базы знаний(БЗ), компонента приобретения знаний, объяснительного компонента и диалогового компонента [2].

Необходимость базы данных обуславливается хранением и использованием данных, которые необходимы для решения задачи в текущий момент времени. Для долгосрочного хранения правил и фактов используется база знаний.

Компонент приобретения знаний автоматизирует наполнение базы знаний правилами и фактами. Объяснительный компонент необходим для того, чтобы раскрыть каким образом и на основании каких данных был получен результат [3]. Для дружественного общения с пользователем как во время решения задачи, так и во время получения знаний используется диалоговый компонент. Решатель состоит из двух механизмов. Интерпретатора, определяющего как использовать данные, находящиеся в базе знаний и диспетчера, определяющего порядок, согласно которому применяются правила. Такие экспертные системы называются статическими (рисунок 1).

Существуют задачи, в которых необходимо учитывать изменение окружающего мира в процессе решения задачи. Для этого к общей модели экспертной системы необходимо добавить подсистему сопряжения с внешним миром, а также подсистему, которая будет моделировать окружающий мир. Такие системы получили название динамических экспертных систем (рисунок 2).

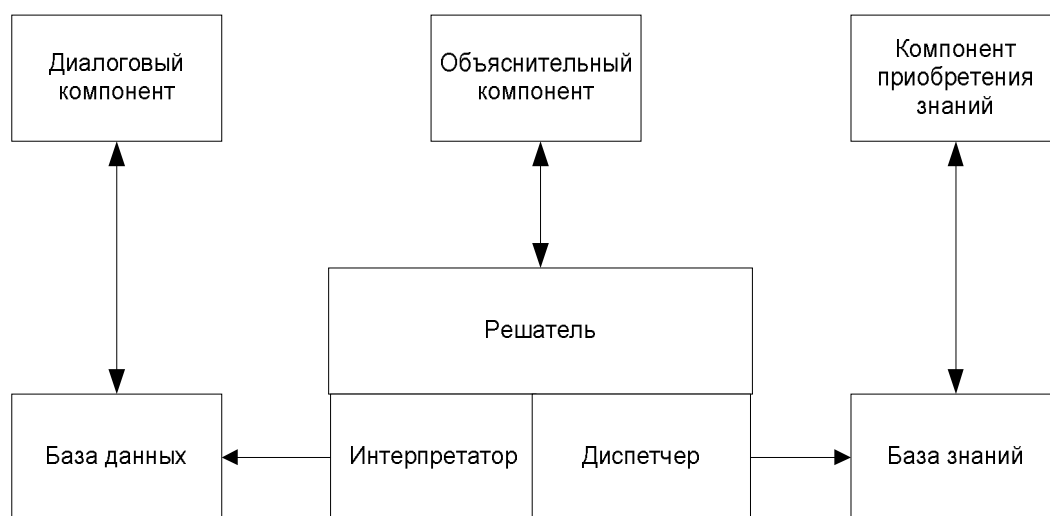


Рис. 1. Статическая экспертная система

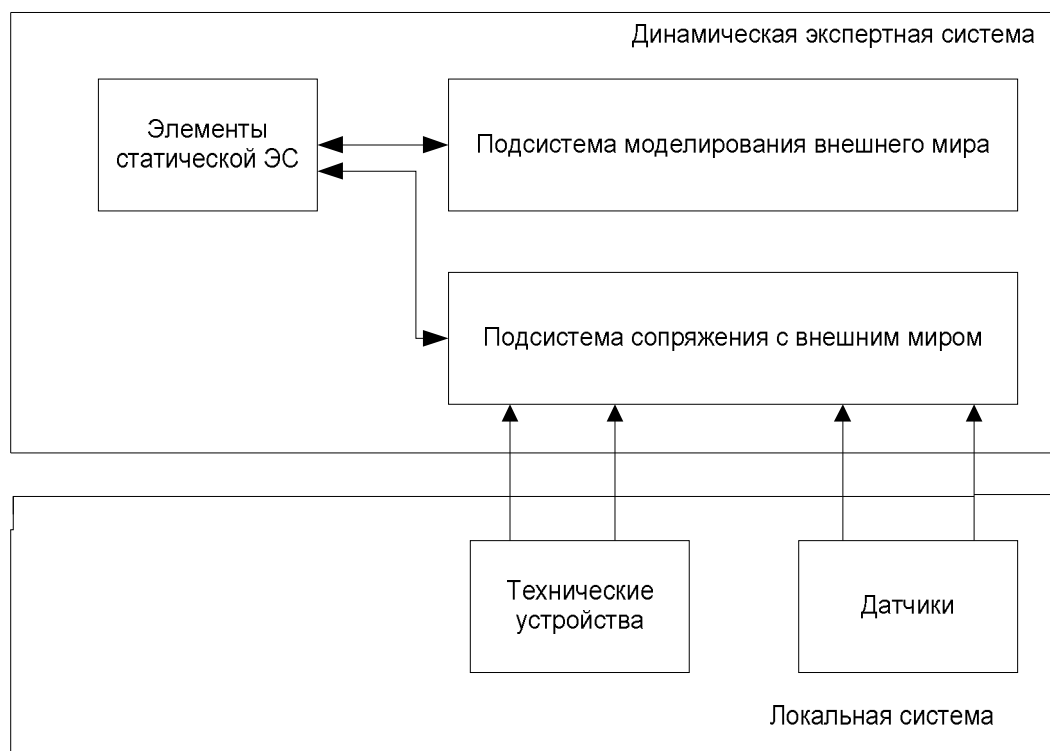


Рис. 2. Динамическая экспертная система

Необходимо разработать экспертную систему для получения ответов на интересующие вопросы студентов и абитуриентов.

Среди программных средств для создания экспертных систем существуют как свободно распространяемые, так и платные. Для создания и эксплуатации экспертной системы в учреждениях высшего образования необходимо использовать программные средства, которые являются свободно распространяемыми.

Такие программные средства так же разделяются по обработке вводимых данных. Они могут быть основаны как на основе нейронных сетей, так и на основе фактов и правил. Для выполнения задачи, поставленной в данной работе будет достаточно использование продукционной экспертной системы.

Рассмотрим следующие программные средства. Первое программное средство, которое необходимо рассмотреть это CLIPS (C Language Integrated Production System). CLIPS является свободно распространяемой (<http://www.ghg.net/clips/CLIPS.html>), продукционной, то есть основанной на фактах и правилах программным средством для разработки экспертных систем.

Язык CLIPS получил большое распространение в государственных организациях и учебных заведениях благодаря мощности, эффективности и переносимости. Например, даже Web-ориентированный инструмент JESS (Java Expert System Shell), использующий язык представления знаний CLIPS, приобрел достаточную известность в настоящее время.

CLIPS содержит объектно-ориентированный язык COOL для создания экспертных систем. Cool представляет собой функциональный строго типизированный язык, то есть проверка типов происходит статически на этапе компиляции. Программа, написанная на языке Cool, включает в себя набор классов. Классы состоят из атрибутов и функций (feature). Все объявленные поля являются видимыми внутри базового и производного классов. Язык построен на выражениях (*expression*). Функции вывода строки возвращают результат в виде экземпляра класса, с помощью которого они были вызваны. Cool содержит также и статические классы String, Int, Bool, от которых нельзя наследоваться и которые не могут принимать значения null (здесь это называется *void*). Объекты классов в данном языке передаются по значению, а не по ссылке.

В CLIPS для создания экспертной системы можно использовать простые типы данных, так же имеются функции для манипуляции с этими данными. Предоставляет конструкторы для пополнения базы фактов и базы знаний. Конструкторы не возвращают никаких значений, в отличие от функций. CLIPS поддерживает загрузку необходимой информации с помощью файла или непосредственного ввода. Есть возможность записи фактов во время диалога с пользователем [4].

Правила в CLIPS состоят из предпосылок и следствия.

```
(deftemplate data (slot a) (slot b))
(defrule multi
  (data (a ? a) (b =(*2 ? a)))
  =>)
(assert (data (a2) (b4)); f-0
        (data (a3) (b9))); f-1
```

Самой распространенной функцией в языке CLIPS является функция assert. Она позволяет добавлять новые факты в список, для удаления факта из списка используется функция retract.

CLIPS так же имеет ряд недостатков. Данной программное средство недостаточно эффективно работает в реальном времени, если необходимо, чтобы реакция была менее 0,1 сек.

OpenCyc – мощное динамическое программное средство для создания экспертной системы, включает в себя онтологическую модель и поддерживает независимые контексты.

Данное программное средство является сокращенным открытым вариантом базы знаний Сус. Данное программное средство позволяет программам решать сложные задачи из области искусственного интеллекта на основе продукционной модели. Интерпретатор и программа, которая позволяет просматривать и изменять базу данных и делать выводы выпущены бесплатно, но только в двоичном виде, без исходных текстов. Она работает как под GNU/Linux, так и под Microsoft Windows.

Для объявления констант необходимо добавить необязательные «#»\$, так же они чувствительны к регистру. Предоставляется возможность для создания констант следующих типов:

- отдельных элементов (individual);
- совокупностей (Collections);
- функций истинности (Truth Functions) которые можно применять к одной или нескольким концепциям и возвращает значение истина или ложь (константы функций истинности начинаются с буквы в нижнем регистре);
- функций Functions (имена констант функций начинаются с буквы в верхнем регистре и заканчиваются строкой «Fn»).

Для описания принадлежности одного класса к другому (подколлекция) используется предикат #\$.genls. Для описания объекта коллекции используется предикат #\$.isa. Предикаты необходимо заключить в скобки перед своими аргументами, например:

(#\$.isa #\$.Apple #\$.Fruit)

означает что «яблоко входит в совокупность фруктов», а высказывание

(#\$.genls #\$.Rose-TheFlowers #\$.Flowers)

читается как «Все розы являются цветами».

Для описания «правил» в OpenCus используется ?, например:

(#\$.implies

(#\$.and

(#\$.isa ?TEMP ?SUBJECT)

(#\$.genls ?SUBJECT?ELEMENTS))

(#\$.isa ?TEMP ?ELEMENTS))

что означает «если TEMP является экземпляром коллекции SUBJECT, а SUBJECT является подколлекцией ELEMENTS, то TEMP является экземпляром коллекции ELEMENTS».

Программное средство OpenCus так же имеет ряд недостатков. Чрезмерная сложность системы – добавления (вручную) данных в систему. Несмотря на своё название система OpenCus не является полностью открытой: данные доступны для редактирования пользователям, но код недоступен для расширения разработчикам.

Для выбора программного средства в качестве среды для разработки экспертной системы предназначенную для абитуриентов и студентов учреждений высшего образования необходимо выделить критерии и провести сравнительную характеристику.

В качестве требований к разрабатываемой системе рассматриваются следующие [5]:

- ответ должен быть получен на языке, близком к естественному;
- возможность задания правил и фактов, относящихся к конкретной предметной области;
- редактирование введенных данных;
- легкая модифицируемость интерфейса (под разные категории пользователей);
- объяснение полученных данных (вывод краткого и длинного вариантов ответа);
- выполнение «рассуждений» асинхронно (обработка по приоритетам поступивших запросов);
- обеспечение наполнения базы;
- защита от несанкционированного доступа.

В соответствии с требованиями к разрабатываемой системе и учитывая все функции и особенности рассмотренных систем была выбрана CLIPS для дальнейшей разработки экспертной системы для получения актуальной и своевременной информации для студентов и абитуриентов Рязанского радиотехнического университета (таблица 1).

Таблица 1. Сравнительная характеристика программных средств для разработки экспертных систем

Программные средства	Возможность задания правил и фактов	Редактирование введенных данных	Возможность изменения интерфейса	Обеспечение наполнения данных	Расширение с помощью дополнительных модулей	Легкость написания правил	Понятность интерфейса
CLIPS	+	+	+	+	+	+/-	+
OpenCyc	+	+	-	+/-	-	+/-	+

В независимости от некоторых недостатков, за экспертными системами будущее. Постепенное совершенствование таких систем неизбежно подтолкнет человека использовать их во всех сферах жизни. Очевидно, компьютер не сможет заменить полностью человека, так, как только человеку удастся находить творческие, нестандартные решения каких-либо проблем, но они значительно облегчат работу эксперта.

Библиографический список

1. Вишневская В.П. Возможности использования экспертных систем в учреждениях высшего образования // Новые информационные технологии в научных исследованиях: материалы XXII Всероссийской научно-технической конференции студентов, молодых ученых и специалистов. Рязанский государственный радиотехнический университет. 2017. С. 58-60.
2. Хабаров С.П. Интеллектуальные информационные системы [электронный ресурс] / Хабаров С.П. – Режим доступа: <http://www.habarov.spb.ru/new/es/index.htm> (дата обращения: 29.09.2017).
3. Каткова А.Л. Экспертные системы [учебно-методическое пособие] / Каткова А.С. – Шадринск, 2011. – 92 с.
4. Дошина А. Д. Экспертная система. Классификация. Обзор существующих экспертных систем // Молодой ученый. — 2016. — №21. — С. 756-758. — URL <https://moluch.ru/archive/125/34485/> (дата обращения: 14.02.2018).
5. Телков А.Ю. Экспертные системы [учебное пособие] / Телков А.Ю. – Воронеж: ИПЦ ВГУ, 2007. – 83 с.

УДК 004.891.2; ГРНТИ 28.23.35

РЕАЛИЗАЦИЯ МЕТОДА ГРАДИЕНТНОГО СПУСКА С ПОМОЩЬЮ БИБЛИОТЕКИ TENSORFLOW НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

В. Мороз

*Рязанский государственный радиотехнический университет,
Россия, Рязань, vlad.moroz62@gmail.com*

Аннотация. В данной работе описаны основные понятия библиотеки TensorFlow, используемой для машинного обучения. Примеры программ приводятся на языке программирования Python. В качестве основного примера приводится реализация алгоритма градиентного спуска.

Ключевые слова. Tensorflow, машинное обучение, градиентный спуск, Python

IMPLEMENTATION OF THE GRADIENT DESCENT METHOD WITH THE TENSORFLOW LIBRARY IN PYTHON PROGRAMMING LANGUAGE

V. Moroz

Ryazan State Radio Engineering University

Russia, Ryazan, vlad.moroz62@gmail.com

Abstract. In this paper, the basic concepts of the TensorFlow library used for machine learning are described. Examples of programs are given in the Python programming language. The implementation of the gradient descent algorithm is given as the main example.

Keywords. Tensorflow, machine learning, gradient descent algorithm, Python.

TensorFlow - это библиотека программного обеспечения с открытым исходным кодом для численного расчета с использованием графов потока данных. Узлы на графе представляют собой математические операции, в то время как ребра графа представляют собой многомерные массивы данных (тензоры), передаваемые между ними. Гибкая архитектура позволяет развернуть вычисления на один или несколько центральных процессоров или графических процессоров на настольном, серверном или мобильном устройстве с помощью единого API.

Для того, чтобы установить библиотеку Tensorflow [1] для Python [2, 3], необходимо выполнить следующую команду в пакетном менеджере pip:

```
pip3 install --upgrade tensorflow
```

Для успешной установки на рабочей машине должен быть установлен Python версии не ниже 3.5. Стоит отметить, что можно установить специальную версию Tensorflow, поддерживающую вычисления на графическом процессоре. Для установки такой версии необходимо выполнить команду:

```
pip3 install --upgrade tensorflow-gpu
```

Дополнительную информацию можно получить на официальном сайте библиотеки - www.tensorflow.org.

TensorFlow был первоначально разработан исследователями и инженерами, работающими в команде Google Brain в исследовательской организации Google Machine Intelligence в целях проведения машинного обучения и исследования глубоких нейронных сетей, но данная система является достаточно гибкой, чтобы быть применимой в самых разных областях.

Чтобы использовать TensorFlow, необходимо создать граф операций, передать в него данные и произвести вычисления. Вершины графа могут содержать либо константы, либо операции (например, сложения или вычитания). Если в Tensorflow выполнить команду вычитания, то сначала система вычислит значения направленных в него ребер графа, а потом произведет указанную операцию.

Инициализация пустого графа выполняется с помощью функции `tf.Graph()`. Однако, при подключении библиотеки пустой граф инициализируется по умолчанию. Таким образом, если граф не инициализируется в начале программы, то будет использоваться граф по умолчанию.

Следующий пример иллюстрирует возможность объявления двух графов на языке программирования Python, в которых будут созданы вершины с двумя разными константами:

```
import tensorflow as tf
Gdef = tf.get_default_graph()
cForGdef = tf.constant(1.53)
G2 = tf.Graph()
with G2.as_default():
    cForG2 = tf.constant(314.2)
```

В переменную `Gdef` записывается граф по умолчанию (пустой). Далее в этот граф (`Gdef`) добавляется вершина, которая является константой. Эта вершина ассоциируется с пе-

переменной `cForGdef`. Далее создается отдельный пустой граф. Ссылкой на этот граф является переменная `G2`.

В TensorFlow есть специальный механизм сессий. С помощью блока `with` в примере открывается сессия, в которой будет использоваться граф `G2` в качестве графа по умолчанию. Таким образом, мы реализуем запись константы 3.14 в граф `G2` и ассоциируем её с переменной `cForG2`. Вне блока `with` в качестве графа по умолчанию будет использоваться граф `Gdef`.

Стоит отметить, что сессии можно открывать так же с помощью вызова функции `tf.Session()`. В качестве результата эта функция возвращает экземпляр данной сессии. Чтобы завершить сессию, необходимо вызвать функцию `close()` данной сессии.

Функция `constant` может принимать несколько параметров:

```
c1 = tf.constant(2.0, shape=[], dtype=tf.float32, name="c1")
```

В данном случае `shape` служит описанием структуры константы `a`. В массиве `shape` указываются размерности этой структуры. Например, если бы параметру `shape` был присвоен массив, состоящий из одного элемента '5', то это бы означало, что константа `c1` представляет собой массив, состоящий из пяти элементов. Если бы параметру `shape` был бы присвоен массив, состоящий из двух элементов '5' и '2', то это бы означало, что константа `c1` является матрицей, состоящей из пяти строк и двух столбцов. В `dtype` записывается тип данной константы.

Для реализации возможности устанавливать входные данные служит специальный тип – заполнители (`placeholders`). Объявляются они следующим образом:

```
p = tf.placeholder(tf.float32, shape=[])
```

Как и в примере с константами, параметр `shape` нужен для установки параметров структуры заполнителя.

Для того, чтобы объявить вершину, которая будет содержать информацию о какой либо операции, необходимо записать следующую запись:

```
mul_result = tf.add(tf.multiply(c1, x), p)
```

В данном случае `mul_result` будет содержать операцию перемножения константы `c1` и переменной `x` с последующим сложением с заполнителем `p`. Стоит отметить, что данную строку можно записать и в более упрощенной форме:

```
mul_result = c1 * x + p
```

Чтобы запустить сессию на выполнение, необходимо выполнить следующую команду:

```
res_mul = session.run([mul_result], feed_dict={p: -5})
```

В первом параметре этой функции передается массив вершин, которые данная функция должна вернуть. В параметр `feed_dict` передается отображение, которое будет соответствовать значениям заполнителя.

Для того, чтобы обучить модель нужна функция потерь и алгоритм оптимизации. Функция потерь принимает значение функции, предсказанной моделью и её фактическое значение, а возвращает разность между ними. В качестве алгоритма оптимизации рассматривается метод градиентного спуска [4, 5].

Ниже приведен пример [6] реализации алгоритма градиентного спуска для функции на интервале от -3 до 4 с 62-ти точками и нормально распределенным шумом. Стоит отметить, что в данной реализации используется библиотека `numpy`.

```
pointsCount = 62 # количество точек
pkgSize = 5 # размер пакета
def f(x):
    return 5*x-2 # оптимизируемая функция
startX = -3 # начальное значение x
```

```

endX = 4 # конечное значение x
sigma = 0.5 # СКО

np.random.seed(0)
d_x = np.arange(startX, endX, (endX-startX)/pointsCount)
np.random.shuffle(d_x) # перемешивание значений массива d_x в случайном порядке
d_y = list(map(f, d_x)) + np.random.normal(0, sigma, pointsCount) # массив значений
функции с использованием шума
placeholder_d_x = tf.placeholder(tf.float32, shape=(pkgSize,)) # заполнитель, с помощью
которого устанавливаются аргументы функции
placeholder_d_y = tf.placeholder(tf.float32, shape=(pkgSize,)) # заполнитель, с помощью
которого устанавливаются значения функции

w = tf.Variable(initial_value=0.1, dtype=tf.float32, name="a")
b = tf.Variable(initial_value=0.0, dtype=tf.float32, name="b")
model = tf.add(tf.multiply(placeholder_d_x, w), b)
lossFn = tf.reduce_mean(tf.square(model-placeholder_d_y)) # функция потерь
optimMethod = tf.train.GradientDescentOptimizer(0.5).minimize(lossFn) # метод
оптимизации

with tf.Session() as session:
    tf.global_variables_initializer().run()
    for i in range(pointsCount//pkgSize):
        feed_dict={placeholder_d_x: d_x[i*pkgSize:(i+1)*pkgSizeSize],
placeholder_d_y: d_y[i*pkgSize:(i+1)*pkgSizeSize]}
        _, l = session.run([optimMethod, lossFn], feed_dict=feed_dict) # запуск оптимизатора
и вычисление потерь
        print("ошибка: %f" % (l,))
        print("a = %f, b = %f" % (w.eval(), b.eval()))
    plt.plot(d_x, list(map(lambda x: wt.eval()*x+b.eval(), d_x)), d_x, d_y, 'ro')

```

В данной реализации размерность входных вершин не совпадают с размерностью переменных. Дело в том, что здесь нужно произвести вычисления над массивами, один из которых имеет лишнее измерение, вычисления производятся отдельно для каждого элемента большего массива, и результатом будет являться массив, имеющий большую размерность. Данный механизм называется пакетным вычислением.

Результатом данной программы будут новые значения переменных *a* и *b*.

Библиографический список

1. Официальный сайт библиотеки TensorFlow [Электронный ресурс], URL: <https://tensorflow.org> (Дата обращения: 28.01.2018)
2. Саммерфилд М. Программирование на Python 3. Полное руководство – М.: Символ плюс, 2009. – 608 с.
3. Raschka S., Mirjalili V. Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow – Packt Publishing, 2015.
4. Карманов В.Г. Математическое программирование. — М.: Наука. Главная редакция физико-математической литературы, 1980. — 252 с.
5. Кричевский М.Л. Интеллектуальные методы в менеджменте – СПб.: Питер, 2005. – 304 с.
6. Введение в машинное обучение с tensorflow [Электронный ресурс], URL: <https://habrahabr.ru/post/326650/> (дата обращения: 02.02.2018).

УДК 004.42; ГРНТИ 20.53.19

РАЗРАБОТКА МОДЕЛИ МНОГОПОТОЧНОГО ГЕНЕТИЧЕСКОГО АЛГОРИТМА ДЛЯ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ

Т.А. Фетисова*Рязанский государственный радиотехнический университет,**Россия, Рязань, pyurova_ta@mail.ru*

Аннотация. В данной работе рассматривается генетический алгоритм и возможность его многопоточной реализации на графических процессорах фирмы NVIDIA с применением технологии CUDA. Также разрабатывается модель алгоритма, описываются основные этапы и элементы распараллеливания данных. Проводится анализ программных средств реализации разработанной модели генетического алгоритма.

Ключевые слова. Генетический алгоритм, многопоточность, технология CUDA, графический процессор, блок, поток.

DEVELOPMENT OF A MODEL FOR MULTITHREADED GENETIC ALGORITHM FOR GRAPHICS PROCESSORS

T.A. Fetisova*Ryazan State Radio Engineering University,**Russia, Ryazan, pyurova_ta@mail.ru*

Abstract. In this paper, the genetic algorithm and the possibility of its multithreaded implementation on graphics processors from NVIDIA are considered with the use of CUDA technology. Develops a model of the algorithm, describes the main stages and the elements of the data parallelization. The analysis of software of realisation of the developed model of genetic algorithm is carried out.

Keywords. Genetic algorithm, multithreading, CUDA technology, GPU, block, thread.

Введение

В настоящее время высокопроизводительные вычислительные машины позволяют за короткое время получать решение больших и сложных оптимизационных задач. Для этого применяют различные алгоритмы обработки данных, среди которых можно выделить эволюционные алгоритмы для поиска решений задач оптимизации высокого уровня сложности, а повышение производительности поиска достигается применением современных компьютерных технологий. Поэтому совместное использование высокопроизводительной техники, параллельных технологий и эволюционных алгоритмов является одним из перспективных направлений исследования оптимизационных задач. К таким эволюционным алгоритмам относятся генетические алгоритмы [5] – мощные независимые методы поиска решения, использующие теорию Дарвина. В генетических алгоритмах учитываются свойства живой природы: отбор (селекция), кроссинговер и мутация для поиска решения в пространстве состояний.

Модель генетического алгоритма на платформе CUDA

Генетический алгоритм начинается со случайной генерации популяции, состоящей из особей (индивидов) с набором хромосом (характеристик), с использованием равномерного закона распределения. На каждой итерации алгоритма для оценки каждой особи вычисляется фитнес-функция (функция полезности). После этого алгоритм заканчивается при условии нахождения подходящего решения или при отсутствии вычислительных ресурсов. В противном случае к особям в популяции применяют операторы мутации и кроссовера (скрещивания) для их видоизменения и дальнейшего анализа. Родителями называют особей из предыдущей популяции, а потомками – особей, полученных с помощью применения эволюционных операторов.

Основная проблема применения генетических алгоритмов заключается в том, что при обработке большого количества решений время выполнения становится ограничивающим

фактором. Однако большинство операций может быть выполнено отдельно для каждой особи популяции различными методами распараллеливания, среди которых выделяют модель «master-slave» («хозяин-рабочий»), островную модель, модель диффузии и гибридную модель.

Островная модель является одним из самых перспективных вариантов параллелизма, поскольку она может достаточно полно использовать вычислительную мощность компьютеров. Согласно данной модели, популяцию разделяют на подпопуляции, называемые островами, которые развиваются независимо друг от друга и свободно позволяют найти различные экстремумы. Также важным оператором является оператор миграции, позволяющим перемешивать хорошие черты, возникающие на островах популяций.

Для распараллеливания генетических алгоритмов существует несколько общецелевых языков высокого уровня как для центрального процессора (ЦП), так и для графических процессоров (ГП). Наибольший интерес представляют ГП, задействующие возможности видеоплаты с помощью технологий CUDA [1] и OpenCL, т.к. разработчикам больше не нужно изучать дополнительные методы программирования через API - интерфейсы. Поэтому ГП целесообразно использовать для ускорения вычислений одинаковых оценочных функций с разными особями, рассматриваемых в качестве разных входных данных функции.

Реализации генетического алгоритма на ГП может быть выполнена на платформе CUDA, т.к. этот инструмент помогает достигнуть высокого ускорения работы алгоритмов. Приложение на CUDA может выполняться на видеоплате NVIDIA GeForce 8 [4] и выше и на любой операционной системе, используя расширение высокоуровневого языка программирования C.

Основным компонентом видеоплаты является память, представленная двумя уровнями — основная (оперативная) память и общая (быстрая) память на микросхеме [2]. Основная память служит связующим звеном между памятью ЦП и ГП, имеет большую емкость и содержит полный набор данных и пользовательских программ. Общая память [3] очень быстрая и имеет ограниченный размер, что является ограничивающим фактором для разработки высокоэффективного генетического алгоритма. Но, несмотря на это, приложения CUDA дают хороший прирост производительности, рационально используя выделенные сегменты памяти.

Модель генетического алгоритма, реализуемого на платформе CUDA, представлена на рисунке 1.

Основная идея распараллеливания заключается в создании островов с применением миграции. Все популяции хранятся в общей памяти на микросхеме в блоках ГП. Каждой особью управляет единственный поток (thread) [6], находящийся в блоке. Это гарантирует интенсивное выполнение операций и мелкозернистый параллелизм для ГП.

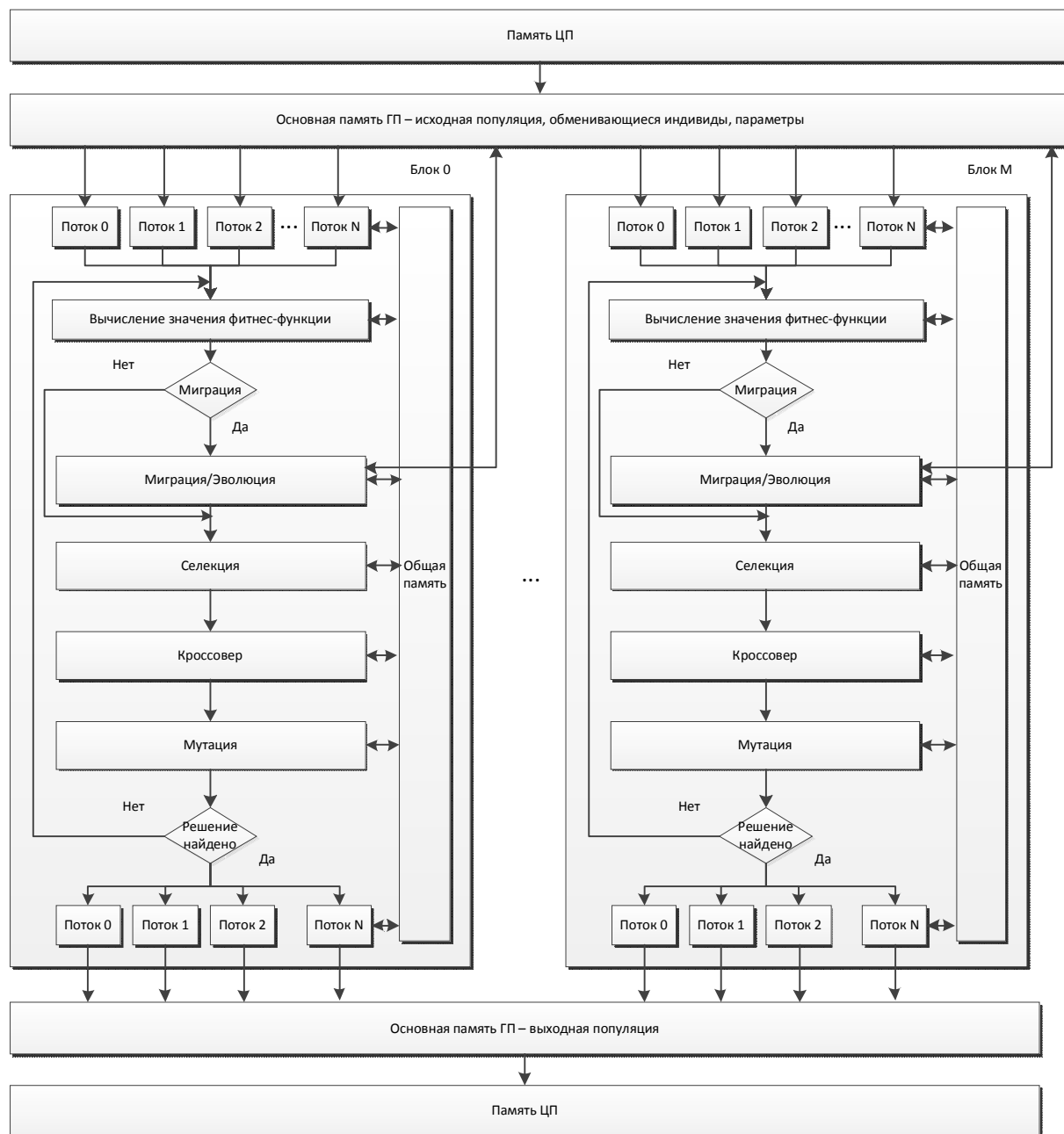


Рис. 1. Модель генетического алгоритма на платформе CUDA

Описание элементов модели генетического алгоритма

Согласно предложенной модели, генетический алгоритм начинается с формирования популяции на стороне ЦП. После этого хромосомы и параметры алгоритма передаются в основную память ГП через системную шину. Затем происходит запуск ядра CUDA, выполняющего вычисление фитнес-функции, миграцию, селекцию, кроссовер, мутацию для GPU. Ядра популяции распределяются на блоки (острова), состоящие из потоков (индивидов). Потоки в каждом блоке читают свои хромосомы из основной памяти в общую память в мультипроцессоре, что позволяет сохранить популяцию острова. После определенное количество поколений эволюционирует отдельно друг от друга, а острова и особи (блоки и потоки) развиваются на ГП параллельно. В каждом поколении имеется функция полезности и происходит селекция индивидов, кроссовер и, при необходимости, мутация.

Чтобы смешать индивиды с разных островов, используют оператор миграции, который требует межостровного взаимодействия через медленную основную память. Сам процесс перемещений особей выполняется асинхронно (без ожидания окончания выполнения предыдущих операций), поскольку синхронизация блоков CUDA сопровождается потерей производительности. Данный подход неприемлем для обычных методов и алгоритмов, требующих согласованности данных, но хорошо работает на стохастических системах, к которым и относится генетический алгоритм.

Потоки (особи) группируются попарно с помощью общих переменных, чтобы оператор кроссовера выполнялся параллельно для всей популяции блока. Каждый индивид из пары случайным образом выбирает одного родителя для вычисления фитнес-функции и выполнения кроссовера. После индекс особи с лучшим значением функции полезности записывается в общую память, чтобы сообщить другому потоку о более подходящем партнере для выполнения скрещивания. Затем параллельно выполняется генерация случайных чисел во всем блоке, а результаты передаются в общий массив. Далее пара потоков ищет свое число из данного массива и сравнивает его с возможностью выполнения кроссовера, т.е. сравнивает с вероятностью наступления данного события, чтобы решить – выполнять скрещивание или нет. Эта подзадача задействует первую половину массива, а вторая половина отводится для выполнения арифметического кроссовера с помощью весов агрегации:

$$O_1 = a * P_1 + (1 - a) * P_2 \quad (1)$$

$$O_2 = a * P_1 + (1 - a) * P_2 \quad (2)$$

где O_1 и O_2 – потомки родителей P_1 и P_2 , отвечающих за веса агрегации a (вероятность выполнения скрещивания).

Процесс миграции осуществляется следующим образом: острова, соединенные однонаправленным кольцом, принимают индивидов от одного соседнего острова. Обмен происходит асинхронно, используя основную память ГП. Локальная популяция блока отсортирована согласно функции полезности. Далее лучшие особи записываются в часть основной памяти, которая принадлежит покинутому блоку, а худшие индивиды записываются как «мигранты» из части основной памяти фактического блока. Операторы сортировки и миграции выполняются отдельно для всех индивидов.

Для практической реализации представленного генетического алгоритма на основе островной модели необходимо применить библиотеку NVIDIA CUDA Toolkit 7.5 и Microsoft Visual Studio 2013. Для проведения полного исследования необходимо использовать видеокарты NVIDIA с разным количеством ядер.

Выводы

Таким образом, генетические алгоритмы широко используются в различных областях человеческой деятельности и наиболее успешно применяются в задачах с большими вычислениями, поэтому применение параллельных технологий для повышения производительности вычислений является актуальным. Генетические алгоритмы предоставляют большое количество материалов для исследований, т.к. имеют много модификаций и параметров, так например, изменение одного из факторов может привести к значительному улучшению результата. Однако применение генетических алгоритмов полезно в общих случаях, т.е. при отсутствии специального алгоритма решения. Тем не менее, на сегодняшний день теоретические оценки эффективности применения параллельных генетических алгоритмов мало исследованы по сравнению с последовательными версиями данных алгоритмов. Использование ГП и технологии CUDA позволит заметно повысить производительность генетических алго-

ритмов за счет параллельного выполнения одинаковых операций алгоритма над индивидами, разбитых на группы для многопоточной обработки.

Библиографический список

1. Bakulev A.V., Bakuleva M.A., Pyurova T.A., Skvortsov S.V. The Implementation on CUDA Platform Parallel Algorithms Sort the Data // Proceedings MECO 2017 - IEEE 6th Mediterranean Conference on Embedded Computing (MECO-2017). 2017. P. 51-54.
2. Nvidia CUDA — неграфические вычисления на графических процессорах [Электронный ресурс] - режим доступа: <http://www.ixbt.com/video3/cuda-1.shtml> (дата обращения 24.01.2018).
3. NVIDIA, C.: Compute Unified Device Architecture Programming Guide. NVIDIA: Santa Clara, CA, 2007.
4. Боресков А.В., Харламов А.А., Марковский Н.Д. Параллельные вычисления на GPU. Архитектура и программная модель CUDA. Учебное пособие. - / Предисл.: В. А. Садовничий. - М.: Издательство Московского университета, 2012. - 336 с.
5. Гладков Л.А., Курейчик В.В., Курейчик В.М. Генетические алгоритмы [Текст]/ Под ред. В.М. Курейчика. — 2-е изд., испр. и доп.— М.: ФИЗМАТЛИТ, 2006. — 320 с. — ISBN 5-9221-0510-8.
6. Скворцов С.В., Пюрова Т.А. Параллельные алгоритмы сортировки данных и их реализация на платформе CUDA // Вестник Рязанского государственного радиотехнического университета. 2016. № 58. С. 42-48.

УДК 004.42; ГРНТИ 50.41.25

ВЗАИМОДЕЙСТВИЕ С ВНЕШНИМИ СЕРВИСАМИ ПОСРЕДСТВОМ API

И. Ширенин

*Рязанский государственный радиотехнический университет,
Россия, Рязань, 387225@gmail.com*

Аннотация. Архитектура и технология взаимодействия с системой-поставщиком посредством выпуска ей API и подключения его системой-потребителем.

Ключевые слова. API, Yandex, удаленный, внешний, система.

INTERACTION WITH EXTERNAL SERVICES USING API

I. Shirenin

*Ryazan State Radio Engineering University,
Russia, Ryazan, 387225@gmail.com*

Abstract. The architecture and technology of interaction with the supplier system through the release of the API and its connection by the consumer system.

Keywords. API, Yandex, remote, external, system.

Постановка задачи

Целью работы является выявление преимуществ использования *API*, описание принципов построения взаимодействия с использованием *API*, а также обзор ранее разработанных взаимодействий.

Введение

API (англ. Application Programming Interface) — это интерфейс программирования приложений. *API* сервиса предоставляет набор готовых процедур, функций и структур, с помощью которых разработчики могут создавать свои программы, приложения, скрипты (далее — приложения) для работы с сервисом. Другими словами, *API* — это конструктор, в котором есть набор деталей (функций, методов) и правил их использования. Из этих деталей вы можете построить собственное приложение [5]. Для выполнения запросов к API вы можете разработать свое приложение на любом языке программирования [4].

Можно считать, что *API* - это некий объект, реализацию которого мы не знаем, однако, можем его использовать. Например, компьютер - объект, реализацию которого знают

очень мало людей, однако, использовать его могут почти все, совершая какие-то действия: просмотр видео, сёрфинг по Интернету, печать текста и прочее. Как это всё работает - мало, кто знает, а вот делать это могут чуть ли не все [6].

При работе через *API* приложение отправляет запрос к сервису и получает ответ, содержащий запрошенные данные. *API* предоставляет возможность автоматизировать процессы, аналогичные вашей работе в веб-интерфейсе.

Преимущества

- Стабильность. Программа совместима с широким спектром модификаций компьютеров.
- Переносимость. Для разных аппаратных платформ может быть написано одинаковое программное обеспечение.
- Простота.

Программные компоненты взаимодействуют друг с другом посредством *API*. При этом обычно компоненты образуют иерархию — высокоуровневые компоненты используют *API* низкоуровневых компонентов, а те, в свою очередь, используют *API* ещё более низкоуровневых компонентов [1].

Функции *API* делятся на 2 направления:

- Возвращающие. На запрос стороннего приложения какого-либо метода с заданными параметрами сервер дает запрашиваемую информацию в определенном формате;
- Изменяющие. Клиент вызывает некоторую функцию сервера, которая вводит новую информацию или изменяет на нем определенные настройки [3].

Обзор *API* взаимодействий

Например, понравившиеся вам видео в Youtube могут автоматически появляться на вашем сайте или в ваших социальных сетях. Это возможно как раз благодаря *API* – когда одна программа использует данные и информацию другой программы. Кстати, встраивание YouTube-видео на свой сайт также возможно благодаря *API* сервиса Youtube.

В общем случае данный механизм применяется с целью объединения работы различных приложений в единую систему. Это достаточно удобно для исполнителей. Ведь в таком случае к другому приложению можно обращаться как к «черному ящику». При этом не имеет значения его внутренний механизм – программист может вообще не знать, что такое *API*.

При работе через *API* приложение отправляет запрос к сервису и получает ответ, содержащий запрошенные данные. *API* предоставляет возможность автоматизировать процессы, аналогичные вашей работе в веб-интерфейсе.

Программист может воспользоваться *API* для получения доступа к функционалу сторонней программы. *API* делает возможным работу ресурсов, которые используют потенциал и мощность другого сайта или программы.

Представьте себе механические часы: сложный механизм из шестеренок и маятников, регулирующий движение стрелок на циферблате. Чтобы вникнуть в работу механизма нужно много времени. Однако повседневное использование часов требует лишь краткого инструктажа: часовая, минутная и секундная стрелки — для определения времени, и колесико — для подвода часов. Если перейти на язык программирования, стрелки и колесико — и есть *API* часов. Он скрывает сложный механизм под набором простых и понятных элементов.

API используется не только для упрощения работы со сложными механизмами. Часто его используют для создания единого принципа работы с разными по своей структуре компонентами. За примером далеко ходить не надо: любая операционная система позволяет нам одинаково легко работать с жесткими дисками, компакт-дисками и флешками. Но ведь «под

капотом» запись и чтение данных с этих носителей требуют выполнения совершенно разных операций. *API* прячет от нас эти различия.

Все мы привыкли к тому, что на разных онлайн-сервисах или платформах мы можем вместо регистрации войти через свои аккаунты в социальных сетях (рисунок 1). Именно это и является использованием *API*, когда сервисы или приложения используют базы данных социальных сетей. При этом сервис может получать информацию о пользователе и использовать ее в своих целях [2].

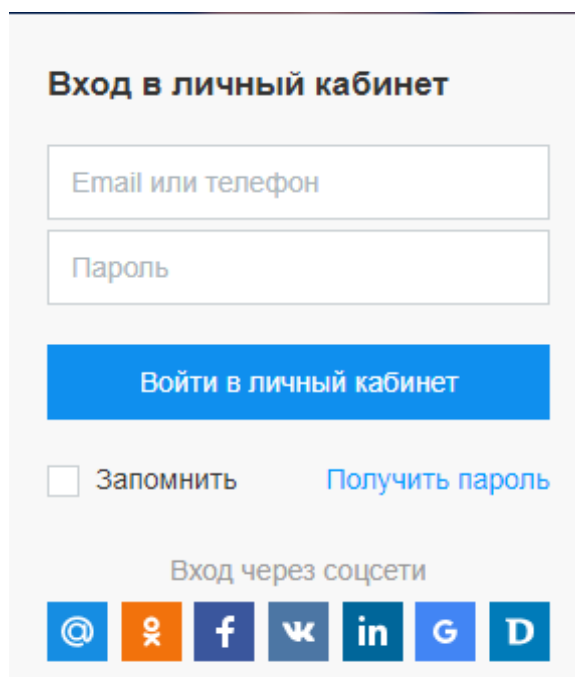


Рис. 1. Использование сторонник API при авторизации.

Построение API взаимодействия

Рассмотрим на примере сервиса *Яндекс.Диск*.

На данный момент API позволяет:

- получать мета-информацию о ресурсах;
- загружать файлы на Диск и скачивать их;
- создавать новые папки;
- копировать, перемещать и удалять ресурсы.

Чтобы пользоваться API необходимо зарегистрировать свое приложение на OAuth сервере. После этого оно сможет запрашивать у пользователя разрешение на доступ к его данным, хранящимся на серверах Яндекса. Яндекс Диск авторизует приложение с помощью OAuth токенов.

Для вызова API необходимо:

- Корректные заголовки *Accept* и *Content-type*: API диска поддерживает только один mime-тип, *application/json*. Любое другое значение приведет к ошибке формата данных.
- *URL*, составленный согласно требованиям к нужному запросу.
- *OAuth* токен, выданный приложению для доступа к диску определенного пользователя. Для запросов, которые не требуют авторизации, это указывается явно.

Для начала необходимо точно понимать, какая из сторон является поставщиком, а какая потребителем. В нашем случае мы будем являться потребителем, а сервис “Диск” поставщиком. Сервис должен выпустить *API*, которое мы подключим к своему проекту. В итоге, нам нет необходимости знать о том, как именно будет реализован какой-либо из поддерживаемых методов работы с Диском, мы будем знать только лишь сигнатуру метода, к которому сможем обратиться, а саму реализацию возьмут на себя разработчики сервиса.

Проще говоря, мы не создаем алгоритм взаимодействия, а пользуемся уже готовым, которые написали за нас разработчики. Схема *API* взаимодействия представлена на рисунке 2.

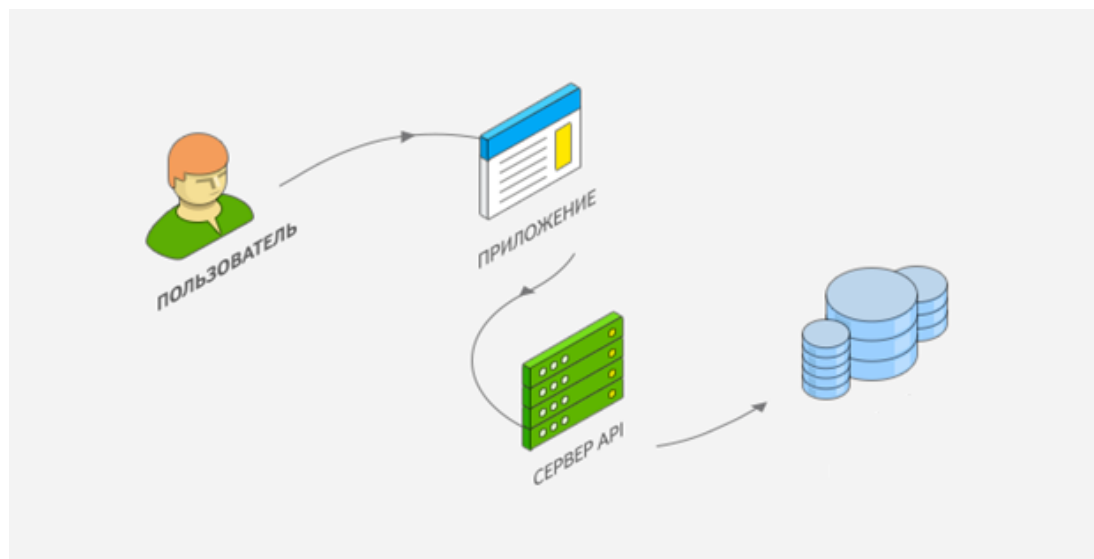


Рис. 2. Иллюстрация взаимодействия с API

Выводы

- Использование *API* сокращает необходимость создавать самостоятельно сложные программы или приложения.
- Использование *API* предполагает использование готовых частей существующих ресурсов, у которых есть доступ к нужной информации и данным.
- Используя *API*, мы застрахованы от внутренних изменений сервиса.

Библиографический список

1. API. [Электронный ресурс], URL: <https://ru.wikipedia.org/wiki/API> (дата обращения 16.01.18).
2. Что же такое API. [Электронный ресурс], URL: <https://te-st.ru/2014/08/15/what-is-api/> (дата обращения 16.01.18).
3. Описание и использование API. [Электронный ресурс], URL: <https://promo.ingate.ru/seo-wikipedia/api/>. (дата обращения 16.01.18).
4. Как выполнить запрос к API. [Электронный ресурс], URL: <https://yandex.ru/adv/edu/direct-api/kak-vypolnit-zapros-k-api> (дата обращения 16.01.18).
5. Что такое API Директа. [Электронный ресурс], URL: <https://yandex.ru/adv/edu/direct-api/chto-takoe-api-direkta> (дата обращения 16.01.18).
6. Что такое API. [Электронный ресурс], URL: <https://myrusakov.ru/php-api.htm> (дата обращения 16.01.18).

УДК 004.42; ГРНТИ 20.53.19

РАЗРАБОТКА АЛГОРИТМА ПОИСКА АССОЦИАТИВНЫХ ПРАВИЛ СЛАБОСТРУКТУРИРОВАННЫХ ДАННЫХ

А.В. Бакулев, М.А. Бакулева

Рязанский государственный радиотехнический университет (РГРТУ)

Российская Федерация, г. Рязань, marina.bakuleva@gmail.com

Аннотация. В данной работе предложен и описан подход к разработке алгоритм поиска ассоциативных правил, основанный на графовой математической модели

Ключевые слова. Ассоциативные правила, добыча знаний, построение псевдобулевой матрицы ассоциаций, производная модельного графа.

DEVELOPMENT OF ALGORITHM FOR FINDING ASSOCIATION RULES FROM SEMI-STRUCTURED DATA

A.V. Bakulev, M. A. Bakuleva

Ryazan State Radioengineering University (RSREU)

Russian Federation, Ryazan, marina.bakuleva@gmail.com

Abstract. This paper devoted to mining association rules. Approach of finding association rules based on graph mathematical model is described. Above all initial data is fuzzy structured.

Keywords. Data mining; association rules, derivative of the modeling graph, implementation of pseudoboollean matrix.

Введение

Задача поиска ассоциативных правил является важной составляющей интеллектуального анализа данных, основанного на современных технологиях Data Mining.

Data Mining предусматривает обнаружение (дословно «добыча» от английского «mining») новых ранее неизвестных знаний, представляющий практический интерес с точки зрения оптимизации процесса принятия решений [1, 2].

Задача поиска ассоциативных правил подразумевает построение импликативных логических конструкций «если, то» на основе анализа большой выборки данных (терабайты информации).

Постановка задачи

Первый алгоритм поиска ассоциативных правил, реализованный в 1993 году, алгоритм AIS, предусматривал поиск кандидатов правило при анализе выборок базы данных. Данный алгоритм не позволял строить ассоциативных правил высокой достоверности. На сегодняшний день самым распространенным является алгоритм Apriori. В основу данного алгоритма заложена идея отсекающих кандидатов в правило не достигающих нижнего порогового значения. При этом данные должны быть представлены как упорядоченный набор значений параметров. Таким образом, если $T = \{t_1, t_2, \dots, t_n\}$ - множество транзакций, $P = \{p_1, p_2, \dots, p_k\}$ - множество значений анализируемого объекта или его параметра, то элементы множества $T = \{t_1, t_2, \dots, t_n\}$ являются псевдобулевыми, то есть принимают значение 0 или 1 в зависимости от того присутствует ли значение P_i в данной транзакции или нет [3].

Для того чтобы сформировать таблицу принадлежности значений к той или иной транзакции, необходимо соблюдать условие структурированности исходных данных. Заметим, что на сегодняшний день актуальной становится задача анализа слабоструктурированной и неструктурированной информации. В частности, большинство данных, содержащихся в глобальной сети, неструктурированы. Поэтому для их анализа и поиска ассоциативных правил, необходимо разработать математическую модель, позволяющую формировать таблицу принадлежности для слабоструктурированных данных.

Описание математической модели

В данной работе предлагается графовая модель для отображения слабоструктурированных анализируемых выборок.

Пусть $M = \{M_1, M_2, \dots, M_n\}$ - множество выборок, в которых могут содержаться данные из множества $V = \{V_1, V_2, \dots, V_k\}$ (рис. 1).

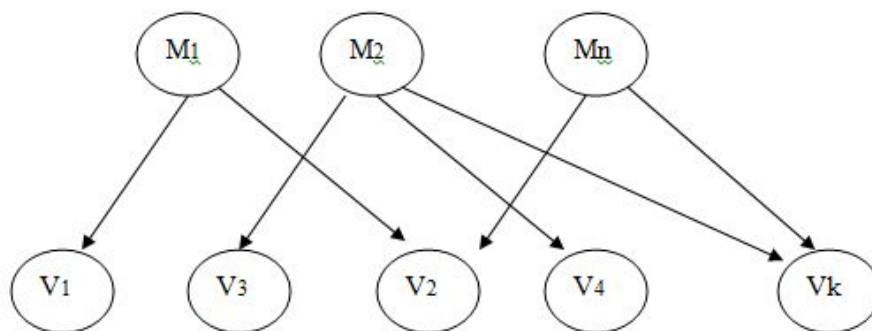


Рис. 1. Графовая модель слабоструктурированных данных анализируемых выборок

Графовую модель можно отобразить в виде матрицы смежности, которая, по сути, будет являться таблицей принадлежности k – анализируемых показателей к каждой из n анализируемых последовательностей

Разработанный алгоритм поиска ассоциативных правил предусматривает нахождение производной $\frac{\partial G}{\partial S}$ графа G .

$$\frac{\partial G}{\partial S} = \frac{f_i - 2 * f_{ij} + f_j}{f_{ij}},$$

где f_i, f_j – собственные частоты дифференцируемых элементов;

f_{ij} – взаимная частота.

Для вычисления производной используется нормализованная матрица смежности графа [4]. Из нормализованной матрицы вычисляется частотная матрица по формуле $F = M^T \times M$

Алгоритм конструирования ассоциативных правил из нормализованной матрицы m_{ij} состоит из следующих шагов:

Шаг 1. Получение частотной матрицы:

$$\text{for } i, j = \overline{1, n} \quad \{ f_{ii} = \sum_i m_{ij}, \quad f_{ij} = \sum_i (m_{ij} \wedge m_{i(j+1)}) \}.$$

Шаг 2. Исследование претендентов в ассоциативное правило:

$$\text{for } q = \overline{1, n^2} \quad \{ \max_q = 0; \quad i_max_q = 0; \quad j_max_q = 0; \}$$

for $i, j = \overline{1, n}$
 {if $(i \neq j) \wedge (f_{ij} > \max)$ then
 { $\max_q = f_{ij}; \quad i_max_q = i; \quad j_max_q = j;$
 $f_{ij} = 0$ } }

Шаг 3. Вычисление производной:

for $q = \overline{1, n^2}$ {
 $der_q = (f_{i_max_q, i_max_q} - 2 * \max_q + f_{j_max_q, j_max_q}) / \max_q,$

Шаг 4. Формулирование ассоциативного правила

for $q = \overline{1, n^2}$ { if $der_q = 0$ then
 { $P_{i_max_q} \rightarrow P_{j_max_q}$ and $P_{j_max_q} \rightarrow P_{i_max_q}$ ассоциативное правило
 }
 else { if $f_{i_max_q, i_max_q} < f_{j_max_q, j_max_q}$ then
 { $P_{i_max_q} \rightarrow P_{j_max_q}$ ассоциативное правило }
 else { $P_{j_max_q} \rightarrow P_{i_max_q}$ ассоциативное правило } } }

Тестирование алгоритма на примере данных (таблица 1) приводит к следующим результатам.

Таблица 1. Экспериментальные данные

Т	зафиксированные наблюдения за состоянием здоровья			
t_1	повышенная температура P_2	учащенный пульс P_3	повышенное давление P_5	избыточный вес P_8
t_2	повышенная температура P_2	учащенный пульс P_3	нормальное давление P_7	нормальный вес P_9
t_3	нормальная температура P_1	слабый пульс P_4	-	нормальный вес P_9
t_4	повышенная температура P_2	учащенный пульс P_3	повышенное давление P_5	-
t_5	нормальная температура P_1	-	пониженное давление P_6	нормальный вес P_9
t_6	повышенная температура P_2	учащенный пульс P_3	пониженное давление P_6	-
t_7	-	слабый пульс P_4	нормальное давление P_7	избыточный вес P_8

Максимальные значения в частотной матрице $\max_1 = 4$, значение производной $der_1 = (4 - 2 * 4 + 4) / 4 = 0$

Следовательно, можно построить ассоциативное правило $P_2 \rightarrow P_3$ («если повышенная температура, то учащенный пульс»). Очевидно, что данное правило является достаточно правдоподобным.

Таким образом, наибольшие значения совместных частот будут соответствовать ассоциациям большей степени достоверности [4].

Вывод

Предложенный алгоритм реализуется в виде модуля программной системы и может быть использован как компонент системы интеллектуального анализа данных.

Библиографический список

1. Корячко В.П., Орешков В.И., Бакулева М.А. Интеллектуальные системы и нечеткая логика. Издательство «Курс». Москва 2017г.
2. Бакулева М.А., Бакулев А.В. Применение вейвлет-преобразования для анализа данных хранилища. Вестник РГРТУ. Научно-технический журнал. Выпуск 21. Рязань: РГРТУ, 2007. С. 57-60.
3. Бакулева М.А., Бакулев А.В., Телков И.А. Алгоритм автоматизации проектирования хранилищ данных Вестник РГРТУ. Научно-технический журнал. Выпуск 23. Рязань: РГРТУ, 2008. С. 90-93.
4. Бакулев А.В., Бакулева М.А. Построение ассоциативных правил на основе дифференцирования графовой модели анализируемой выборки // Вестник Рязанского государственного радиотехнического университета. 2013. № 46-2. С. 86-89.

УДК 004.4; ГРНТИ 50.43

ПОСТРОЕНИЕ СРЕДСТВА ТЕСТИРОВАНИЯ ОБУЧАЮЩИХСЯ НА ОСНОВЕ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ БАЗ ЗНАНИЙ

В.В. Миронов

*Рязанский государственный радиотехнический университет,
Россия, Рязань, mironov.v.v@rsreu.ru*

Аннотация: в статье рассматриваются основные принципы проектирования средства тестирования обучающихся на основе системы автоматизированного проектирования баз знаний.

Ключевые слова. Проектирование, формообразование, база знаний, морфология

CREATION OF MEANS OF TESTING OF THE KNOWLEDGE BASES STUDYING AT THE BASIS OF THE COMPUTER-AIDED ENGINEERING SYSTEM

V.V. Mironov

*Ryazan State Radio Engineering University,
Russia, Ryazan, mironov.v.v@rsreu.ru*

Abstract: in article the basic principles of design of means of testing of the knowledge bases studying at a basis of the computer-aided engineering system are considered.

Keywords: design, shaping, knowledge base, morphology

Введение

Обучение иностранных студентов русскому языку обязательно включает в себя обучение их правилам формообразования (также иногда называемым правилами словоизменения) [2], поскольку формообразование в русском языке является весьма сложным [6].

Существенную помощь в подготовке практических занятий может принести подготовка обширного материала для проверки знаний, поскольку простая тренировка с тестирующей программой, содержащей даже только наиболее часто употребляемые слова, помогает запомнить правильное их словоизменение [5].

Для построения такой системы наиболее эффективным является применение обобщённого морфологического модуля, позволяющего формировать материал для тестов с наименьшими трудозатратами [1].

Поэтому в качестве базы для создания программного модуля контроля знаний по формообразованию русского языка была использована система проектирования баз знаний по формообразованию естественных языков *Salvinia*.

Эта система позволяет в общем виде описать правила словоизменения для любого языка с постфиксным формообразованием (в том числе русского), содержит средства автоматизации процесса пополнения словаря, включающие:

- импорт групп слов, изменяющихся по одинаковым правилам,
- автоматизированное определение наиболее вероятного набора правил словоизменения для неизвестного слова (рис. 1),
- автоматическую генерацию полных парадигм.

Реализация

Ядро системы может быть применено в прикладных программах независимо от самого средства редактирования баз данных, и вполне подходит в качестве основы для создания системы тестирования обучающихся.

База знаний о формообразовании конкретного языка (в данном случае русского) представляет собой XML-документ, хранящий как правила формообразования, так и сами слова, сгруппированные в соответствии с заданной моделью формообразования языка по частям речи и постоянным грамматическим признакам [2].

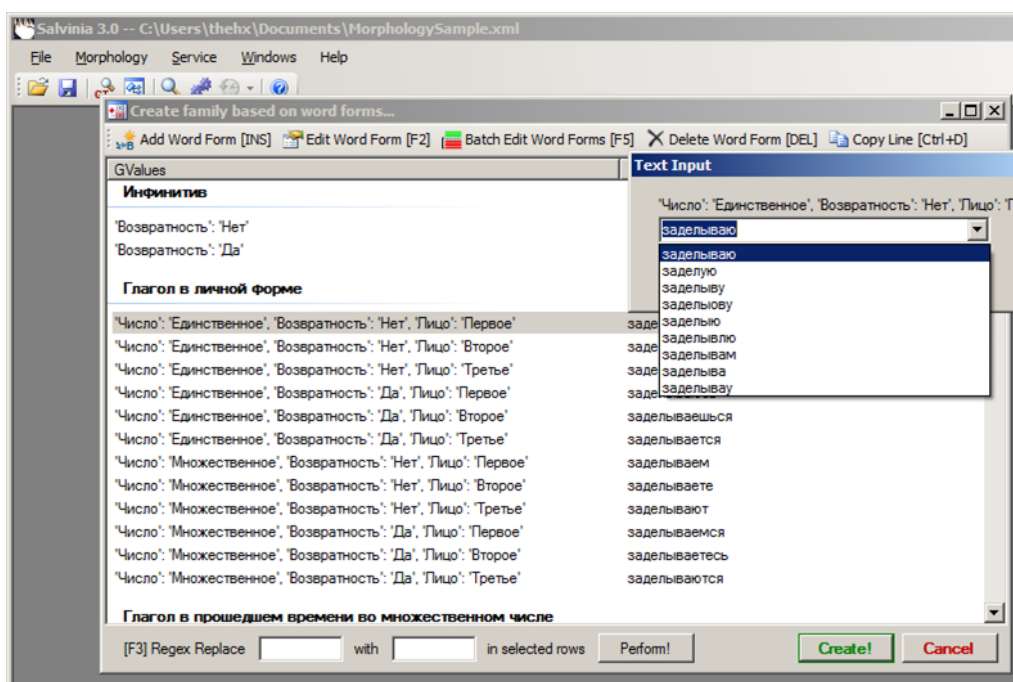


Рис. 1. Автоматическое определение форм неизвестного слова

При создании базы знаний о формообразовании конкретного языка с помощью системы автоматизированного проектирования баз знаний *Salvinia* [3] инженер по знаниям строит схемы формообразования для всех частей речи данного языка. Пример таковой схемы для русских глаголов приведён на рисунке 2.

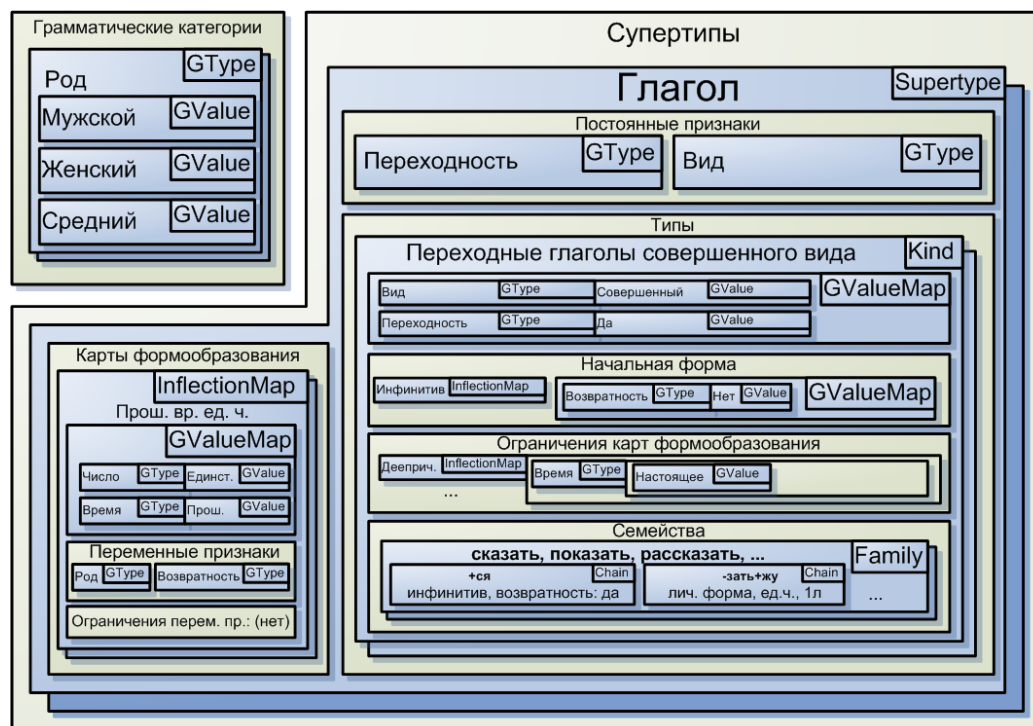


Рис. 2. Структура знаний о словоизменении русских глаголов в системе Salvinia

Структурные блоки, их назначение, точный состав и правила композиции детально описаны в [4].

Созданная система контроля знаний по морфологии Salvinia Examiner является вспомогательным средством для обучения морфологии русского языка. Этот комплекс обучающего и контролирующего программного обеспечения позволяет решать следующие задачи в обучении:

- формирование умения определять грамматическую информацию для написанных слов (распознавать число и падеж имён существительных, время глагола и т.д.);
- формирование навыка построения требуемых форм слов (например, умения просклонять некоторое существительное).

Краткое описание системы

Система контроля знаний Salvinia Examiner включает в себя следующие компоненты:

- Salvinia Editor – система автоматизированного проектирования баз знаний о морфологии языка. В этом приложении инженер по знаниям (преподаватель) может расширять базу знаний об изучаемом языке (рис. 1, 3);
- Salvinia Test Editor – редактор заданий для проведения тестирования (рис. 3) – позволяет составлять файлы заданий, в рамках которых приложение-тестер проводит сеансы тестирования людей, изучающих морфологию данного языка;

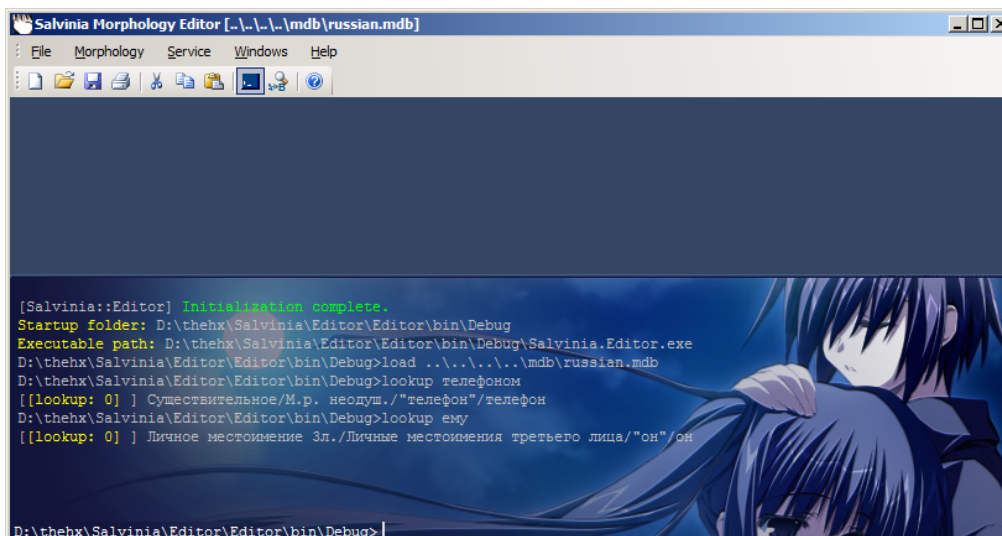


Рис.3. Поиск слов в базе знаний из служебной командной строки системы проектирования

- Salvinia Tester (рис. 4, 5, 6, 7) – средство проверки знаний, работающее как в режиме обучения (с подсказками), так и в режиме контроля, выполняющее проверку правильности ответов, ведущее подсчёт затраченного времени и составляющее полный протокол тестирования;

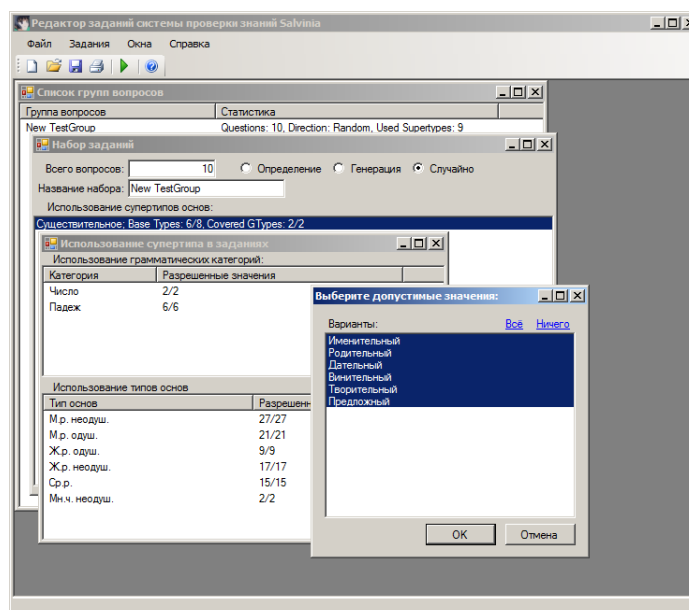


Рис.4. Редактирование состава тестовых заданий

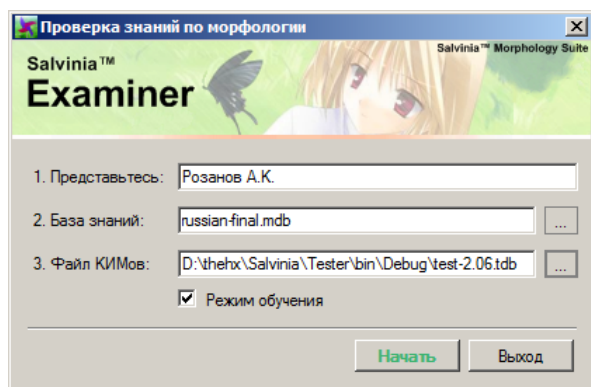


Рис.5. Вход в систему проверки знаний

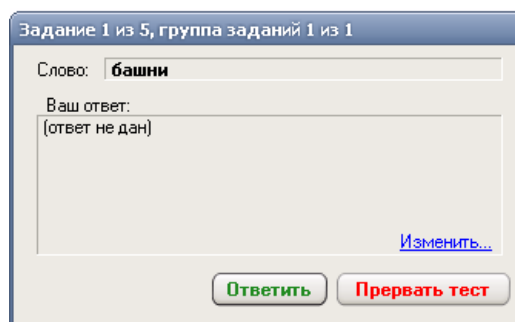


Рис.6. Задание на определение формы слова

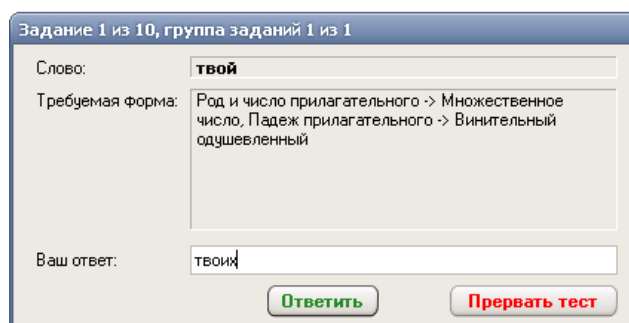


Рис.7. Задание на генерацию формы слова

- Salvinia Log Viewer (рис. 8,9) – средство просмотра протоколов тестирования.

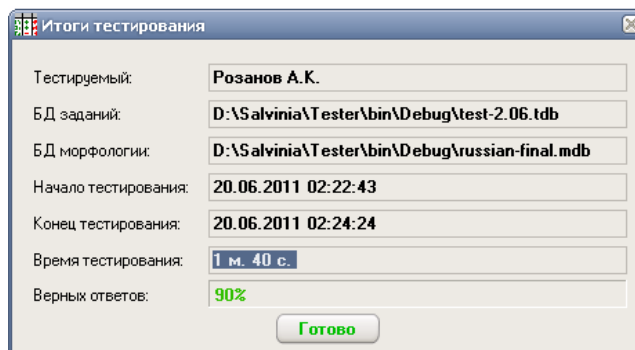


Рис. 8. Результаты тестирования

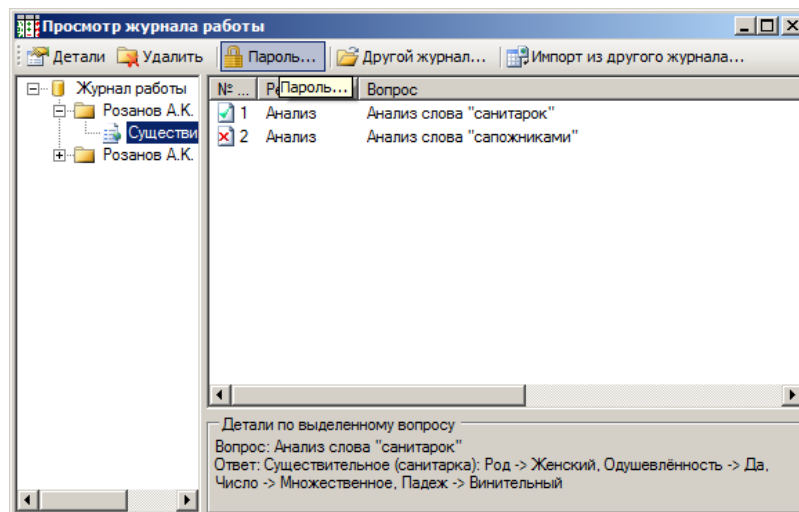


Рис. 9. Протокол проведённого сеанса тестирования

Работа с системой

Последовательность действий при прохождении теста имеет вид:

- 1) ввод информации о тестируемом и подключение к базе знаний морфологии и базе заданий;
- 2) последовательный выбор или ввод ответов на предлагаемые вопросы;
- 3) внесение информации о тестировании в журнал событий и вывод результатов тестирования на экран.

Система контроля знаний работает в двух режимах: режим обучения (ответы анализируются, выводится справка и правильный ответ) и режим контроля (выводится только окно статистики). По окончании тестирования система переходит в изначальное состояние, и тестирование может быть начато заново. Модуль тестирования открывает базу заданий и для каждой из групп запускает цикл на требуемое число повторений (указанное как «число вопросов» в группе заданий) генерации вопроса и проверки введенного или выбранного пользователем ответа. Логика проверки ответа приведена ниже.

1) В случае задачи определения формы слова указанная пользователем трактовка сравнивается со всеми допустимыми трактовками, которые предоставляются морфологическим ядром *Salvinia Engine* исходя из данных активной базы знаний о формообразовании, и если (и только если) пользователь ввёл подходящий вариант трактовки словоформы, и этот вариант входит в подмножество, заданное текущей базой тестов, ответ пользователя считается верным;

2) В случае задачи генерации формы слова морфологическое ядро из данной основы слова всеми возможными способами (единственность способа необязательна, например, прилагательное «красный» имеет формы «красною» и «красной» для творительного падежа женского рода) строит словоформы, имеющие требуемую грамматическую информацию. Системе контроля знаний остаётся только проверить, совпадает ли ответ пользователя хотя бы с одной из полученных словоформ, – тогда, и только тогда он считается верным.

Пользователь в любой момент может остановить тестирование, нажав на соответствующую кнопку. Все вопросы, на которые ещё не был дан ответ, в этом случае не фиксируются в журнале событий. Вместе с тем, они не считаются правильными, и доля верных ответов по-прежнему вычисляется как отношение данных верных ответов к запланированному общему числу вопросов.

Выводы

В статье было описано построение средства тестирования обучающихся на основе системы автоматизированного проектирования баз знаний, обозначена практическая ценность проектируемого продукта, кратко описана реализация работы системы, обозначены компоненты, описаны их действия.

Продукт тестирования ориентирован, прежде всего, на обучение иностранных студентов русскому языку, а именно правилам формообразования, словоизменения, а также оказывает существенную помощь в подготовке практических занятий.

Библиографический список

1. Миронов В.В., Розанов А.К., Нелюхин С.А. Системные методы и компьютерные технологии в учебном процессе /под ред. проф. Миронова В.В. Рязань: Book Jet, 2017. – 90 с.
2. Миронов В.В., Заволокин А.И., Розанов А.К. Обучение студентов русско-английскому и англо-русскому переводу с автоматизированным контролем знаний через применение компьютерных программ, реализующих алгоритмы активной и пассивной грамматик // *Cloud of Science*. – 2017. – Т. 4. - №1 – 52 с.
3. Миронов В.В., Гришина В.В., Заволокин А.И., Розанов А.К., Нелюхин С.А. Современные технологии в процессе обучения. Рязань: Book Jet, 2017. – 162 с.

4. Розанов А.К. Организация словаря в системах генерации и определения форм слов естественных языков // Вестник Рязанского государственного радиотехнического университета, № 49, 2014. С. 55-63.
5. Розанов А.К. Метод предсинтаксического анализа текста на основе знаний о формообразовании естественного языка // Математические методы в технике и технологиях. Материалы XXVIII международной научной конференции / Рязань, 2015.
6. Пруцков А.В., Розанов А.К. Методы морфологической обработки текстов // Прикаспийский журнал: управление и высокие технологии, № 3 (27), 2014. С. 119-133.

УДК 519.252; ГРНТИ 50.43.15

МЕТОДОЛОГИЧЕСКИЕ ОСНОВЫ ПОСТРОЕНИЯ ПРОГРАММНОГО КОМПЛЕКСА СИНТЕЗА ПРОЕКТНЫХ РЕШЕНИЙ В ЗАДАЧАХ ВНЕДРЕНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ РАСЧЕТОВ С НАСЕЛЕНИЕМ ЗА ЖИЛИЩНО-КОММУНАЛЬНЫЕ УСЛУГИ

С.В. Аникеев*, В.В. Миронов**

** Рязанский государственный радиотехнический университет,
Россия, Рязань, s.anikeev@abonentplus.ru*

*** Рязанский государственный радиотехнический университет,
Россия, Рязань, mironov.v.v@rsreu.ru*

Аннотация. Рассматриваются основы построения программного комплекса синтеза проектных решений в задачах внедрения информационных систем расчетов с населением за жилищно-коммунальные услуги.

Ключевые слова. Система, проектирование, программный комплекс, информационная система

METHODOLOGICAL BASES OF CREATION OF THE PROGRAM COMPLEX OF SYNTHESIS OF DESIGN DECISIONS IN PROBLEMS OF INTRODUCTION OF INFORMATION SYSTEMS OF SETTLEMENTS WITH THE POPULATION FOR HOUSING AND COMMUNAL SERVICES

S.V.Anikeev*, V.V.Mironov**

*Ryazan State Radio Engineering University,
Russia, Ryazan, s.anikeev@abonentplus.ru*

*** Ryazan State Radio Engineering University,
Russia, Ryazan, mironov.v.v@rsreu.ru*

Abstract. Basics of creation of a program complex of synthesis of design decisions in problems of introduction of information systems of settlements with the population for housing and communal services are covered.

Keywords. System, design, program complex, information system

Традиционно задачи расчетов с населением за жилищно-коммунальные услуги (ЖКУ) в сфере ЖКХ решаются с использованием средств вычислительной техники и специализированного прикладного программного обеспечения - расчетных систем. В силу значительных объемов обрабатываемой информации и широких функциональных возможностей, системы данного типа относят к числу корпоративных информационных систем [1].

К числу наиболее характерных задач, решаемых такого рода системами, можно отнести следующие [2].

1) Расчет и перерасчет объемов и стоимости ЖКУ в разрезе потребителей, поставщиков, режимов потребления услуги, узлов транспортной сети с точностью до одного дня, а в случаях недопоставки услуг - с точностью до часа.

2) Регистрация и учет информации о платежах за ЖКУ, поступающую через сеть территориально-распределенных пунктов приема платежей, включая сети терминалов самообслуживания, Интернет-сайты, мобильные устройства.

3) Информационная поддержка комплекса задач, связанных с эксплуатацией и обслуживанием транспортных сетей, сетей потребляющего и контрольно-измерительного оборудования.

4) Разработка, формирование и печать комплектов выходных документов - счетов, извещений, отчетов, справок и т.д.

В целом системы подобного рода характеризуются:

- большими объемами обрабатываемой информации;
- большим количеством одновременно работающих пользователей;
- высокими требованиями к скорости обработки информации, стабильности и качеству работы;
- интенсивными потоками информационного взаимодействия со смежными региональными и государственными информационными системами сферы ЖКХ.

Приведенные характеристики объективно формируют для предприятий ЖКХ потребность в использовании информационной системы, отсутствии которой делает их работу в соответствующей сфере фактически невозможной [3]. Более того, от качества решения этих задач и, следовательно, качества программной системы зависит экономическая эффективность работы предприятия, его успешность в конкурентной борьбе. Все это подталкивает предприятия к поиску и внедрению новых, более эффективных информационных систем.

Существенным, однако, является то, что даже при наличии на рынке более эффективной системы и потенциального желания предприятия ЖКХ провести модернизацию своего программного обеспечения сам процесс обновления является сложной организационно-технической задачей, включающий множество рисков, реализация даже части которых способна нанести непоправимый урон репутации предприятия.

Минимизировать риски, сделать процесс смены информационной системы управляемым, стабильным и предсказуемым может только тщательное проектирование процесса внедрения информационной системы.

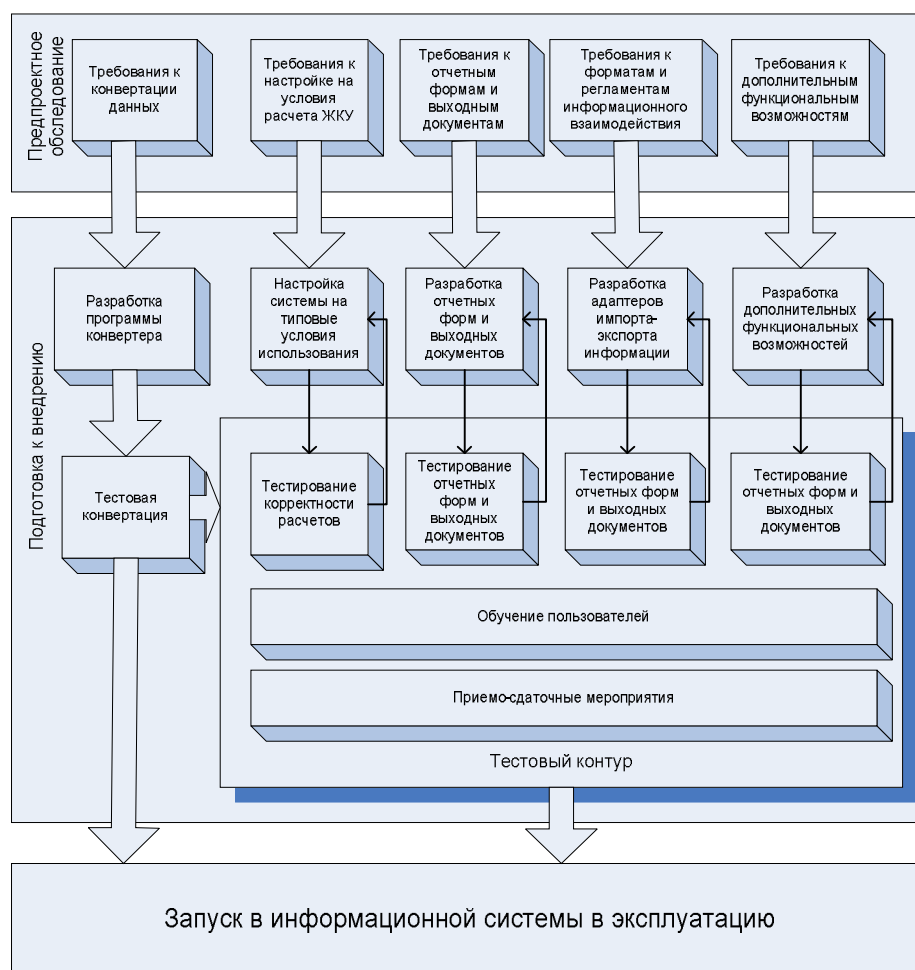


Рис 1. Структурная схема процесса внедрения информационной системы расчетов с населением за ЖКУ

Опираясь на многолетний опыт выполнения проектов в данной сфере, авторы выделяют следующие его этапы (рис.1).

- Предпроектное обследование предприятия. Разработка регламентов информационного обмена. Разработка регламентов формирования типовых отчетных форм.
- Анализ структур данных, предоставленных для конвертации.
- Разработка программы-конвертера. Предварительная конвертация. Сверка данных.
- Настройка системы на условия расчета. Реализация алгоритмов и методик расчета за ЖКУ, специфичных для данного предприятия.
- Разработка дополнительного функционала системы.
- Разработка типовых адаптеров информационного обмена (онлайн, оффлайн).
- Обучение пользователей.
- Пусконаладочные работы.
- Этап опытной эксплуатации.
- Финальная конвертация.
- Начало промышленной эксплуатации. Продуктивная поддержка системы после внедрения.
- Сопровождение.

Приведенная сложная многоуровневая схема процесса внедрения, высокие требования к качеству и срокам решения задач требуют разработки программного комплекса синтеза

проектных решений в задачах внедрения информационных систем расчетов и учета платежей населения за жилищно-коммунальные услуги на предприятиях сферы ЖКХ [4].

Основными структурными единицами комплекса являются.

- Модуль управления требованиями к конвертации данных.
- Модуль управления требованиями к типовым отчетным формам.
- Модуль управления требованиями к форматам и регламентам информационного взаимодействия с внешними информационными системами.
- Модуль управления требованиями к настройке на условия расчета за ЖКУ.
- Модуль управления требованиями на расширение функциональных возможностей системы.
- Модуль регистрации обращений и технической поддержки пользователей.

Модули управления разработаны на основе общих идей теории управления, развитые в работах [5, 6, 7].

Согласованное использование модулей комплекса позволяет синтезировать проект внедрения системы расчетов за ЖКУ с населением на предприятиях сферы ЖКХ.

Полная автоматизация подготовки проектной документации на всех этапах позволяет существенно сократить количество ошибок, обеспечить функциональную полноту процесса внедрения, выявить и взять под контроль организационные и технические риски. Предварительное нормирование этапов работ позволяет проводить финансово-стоимостной анализ выполняемого проекта.

Выводы. В статье рассмотрены методологические основы построения программного комплекса синтеза проектных решений в задачах внедрения информационных систем расчетов с населением за жилищно-коммунальные услуги. Обозначены этапы процесса внедрения информационной системы. Перечислены основные структурные единицы разработки программного комплекса синтеза проектных решений в задачах внедрения информационных систем расчетов и учета платежей населения за жилищно-коммунальные услуги на предприятиях сферы ЖКХ.

Библиографический список

1. Интернет-ресурс <https://e-educ.ru/ism14.html> Дата просмотра: 16.03.2018
2. Интернет-ресурс: <https://sites.google.com/site/korpinfsis/home/osnovy-i-osnovnye-ponatiia-korporacii-i-kis> Дата просмотра: 15.03.2018
3. Интернет-ресурс: <https://www.iemag.ru/analytics/detail.php?ID=30294> Дата просмотра: 11.01.2018
4. Интернет-ресурс: [http://www.tadviser.ru/index.php/Проект:ГИС_ЖКХ_\(Государственная_информационная_система_жилищно-коммунального_хозяйства\)](http://www.tadviser.ru/index.php/Проект:ГИС_ЖКХ_(Государственная_информационная_система_жилищно-коммунального_хозяйства)) Дата просмотра: 20.12.2017
5. Миронов В.В., Северцев Н.А. Методы анализа устойчивости систем и управляемости движением. / ВЦ РАН.- М.: Изд-во РУДН, 2002. 165 с.
6. Миронов В.В., Митрохин Ю.С. Задача об управляемости систем оптимального регулирования и методы ее решения // Вестник РГРТА: Рязань, 2000. Вып. 7. С. 57-64.
7. Миронов В.В. Фундаментальные математические проблемы. Комментарии (Пленарный доклад) / XXVIII межд. науч. конф. «Математические методы в технике и технологиях». – Рязань: РГРТУ, 2015. С. 3-12.

УДК 681.5 ГРНТИ 50.43.19

ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ СИСТЕМ ВИЗУАЛИЗАЦИИ ИЗОБРАЖЕНИЙ ДЛЯ ЛЕТАТЕЛЬНЫХ АППАРАТОВ

А. Братулин, А. Киселев, М. Никифоров

*Рязанский государственный радиотехнический университет,
Россия, Рязань*

Аннотация. В данной работе описаны основные этапы проектирования, разработки и сертификации оборудования, используемые для внедрения в системы воздушных судов.

Ключевые слова. Бортовая аппаратура, сертификация.

EVALUATION OF ECONOMIC EFFICIENCY OF VISUALIZATION SYSTEMS FOR AIRCRAFT APPARATUS SYSTEMS

A. Bratulin, A. Kiselev, M. Nikiforov

*Ryazan State Radio Engineering University,
Russia, Ryazan*

Abstract. This work describes the main stages of design, development and certification of equipment used for deployment in aircraft systems.

Keywords. Airborne equipment, certification.

Введение

Основной целью систем безопасности систем визуализации летательных аппаратов прогнозирование, оценка и анализ технических рисков в общем процессе оценки стоимости жизненного цикла авиационного оборудования. Экономическая эффективность выражается в сокращении сроков разработки новых образцов и обеспечении максимальной коммерческой эффективности при обусловленном выполнении требований безопасной эксплуатации оборудования.

В данной статье будут рассмотрены этапы и методы разработки оборудования на предмет возможности сокращения итераций в разработке систем летательных аппаратов.

Этапы разработки оборудования

В проектной деятельности присутствуют следующие этапы оценки надежности:

1. Выявление целей в сфере надежности
2. Анализ актуальности и значимости работ в сфере надежности
3. Создание плана действий для достижения целей в сфере надежности
4. Выполнение соответствующих действий в сфере надежности
5. Анализ результатов осуществленных действий
6. Оценка достигнутых результатов для дальнейших действий по усовершенствованию

В текущий момент продвижение менеджмента качества характеризуется переносом основных акцентов с оценки готовой продукции на оценку и менеджмент процессов ее создания.

Анализ зарубежного опыта предоставил следующую информацию[2]:

- Сертификация является ключевым средством гарантии качества
- Методы управления качеством рациональны только тогда, когда они осуществляются на всех этапах разработки

Основным показателем, обеспечивающим гарантию проектирования, является системный подход к проектированию бортовой аппаратуры воздушных судов (ВС). Под гарантией проектирования понимается процесс, состоящий из специально запланированных систематических мероприятий, обеспечивающих в совокупности уверенность в том, что ошибки или упущения в требованиях или проекте выявлены и устранены таким образом, что реализованная система будет удовлетворять сертификационным требованиям.

При решении перечисленных задач математическое моделирование используется, когда известно достаточно достоверное математическое описание моделируемого процесса. Полу натурное моделирование применяется для оценки аппаратных решений, эргономической оценки и при отработке образцов для уточнения технических решений, получения объективных оценок для принятия решений о проведении натурных испытаний и возможности пере хода от одного этапа испытаний к последующему.

Значительный интерес представляет задача обоснования достаточности обработанности системы на различных этапах ее разработки и возможности перехода на следующий в технологической цепочке этап.

Эта задача решается с использованием так называемых «кривых роста надежности».

На каждом этапе разработки системы происходит обнаружение и устранение различных дефектов, т. е. корректировка и доработка системы. При этом наблюдается улучшение характеристик системы. Определив отказ как любое несоответствие параметров и характеристик системы предъявляемым требованиям, и учитывая вероятностный характер процессов обнаружения дефектов, получим возможность использования для описания этого процесса хорошо известного и достаточно полно разработанного математического аппарата теории надежности.

Аналогом отказа, в данном случае, является обнаружение дефекта. Вероятность события при котором система будет работать без отказов называют надежностью системы.

При тестировании системы ее безотказность растет. Данную закономерность можно описать экспоненциально, отталкиваясь от вышеупомянутого определения надежности системы (R)[4].

$$R_i(t_i) = a_i - (a_i - R_{oi})\exp\{-Q_i t_i\},$$

где R_i - этап разработки;

t_i – разработка на данном этапе во временном выражении;

R_{oi} – нулевое значение надежности (начальное);

Q_i – интенсивность обнаружения, а так же исправления дефектов на данном этапе;

a_i – маргинальное (предельное) значение надежности для данного этапа, определяемое цельностью имитации условий эксплуатации системы.

Использование экспоненциальной модели обусловлено тем, что рост надежности систем пропорционален обнаруженной ненадежности, т. е.

$$R = Q(a - R),$$

а отличие параметра «а» от единицы определяет проявление функционировании любой системы случайных отказов, не требующих переработки системы.

При исследовании динамики роста надежности различных летательных систем в процессе их экспериментальной обработки было выявлены следующие закономерности:

– интенсивность обнаружения дефектов уменьшается с переходом на следующие этапы, т. е.

$$Q_{i-1} > Q_i,$$

что объясняется как уменьшением числа неопределенных дефектов, так и возрастанием сложности в условиях их обнаружения;

– маргинальное (предельное) значение растет с переходом на следующий этап, т. е. $a_{i-1} < a_i$,

что объясняется все наиболее полной имитацией реальных эксплуатационных условий при переходе от математического моделирования к летным испытаниям.

В результате может быть поставлена и решена задача оптимального по времени поэтапного перехода, а именно:

$$T = t_{i-1} + t_i = \frac{1}{Q_{i-1}} \ln \frac{a_{i-1} - R_{oi-1}}{a_{i-1} - R_{oi}} + \frac{1}{Q_i} \ln \frac{a_i - R_{oi}}{a_i - R_{oi+1}} = \min R_{oi}$$

Решением для этой оптимизационной задачи состоит условие равенства возрастания надежности на $i-1$ -ом и i -ом этапах отработки в точке перехода R_{oi} , а именно

$$Q_i(a_i - R_{oi}) = Q_{i-1}(a_{i-1} - R_{oi}).$$

Если изменить время на цену отработки на разных этапах, то получится аналогичное условие.

Анализируя зарубежные технологии, можно заметить, что важнейшую роль в отечественной оценке определения соответствия главных показателей летной пригодности самолетов играют летные испытания, в то время как зарубежом около 80% всех проблемных ситуаций, связанных с разработкой авиационного оборудования, решаются на земле[3].

Каждая система может разрабатываться сразу на нескольких этапах проектирования. Процесс разработки компонентов системы может включать в себя несколько жизненных циклов.

Концептуальное проектирование является первым этапом в типовом подходе к проектированию систем ВС. Обычным для данной системы является концепция «сверху вниз». В начале данного процесса проектирования находятся требуемые функции для ВС. Заканчивается проектное проектирование сертификацией.

Каждый элемент (компонент) системы создается с учетом всех требований функциональности и безопасности по четким критериям, установленным архитектурой данной системы. Вариантов реализации архитектуры системы может быть несколько.

Сроки реализации, затраты на производство, используемые технологии и готовность промышленности могут вносить дополнительные ограничения в процесс реализации.

Распределение требований между программным обеспечением и аппаратной частью вместе с проектированием архитектуры системы являются итеративными процессами, которые тесно связаны между собой. Каждый цикл системы повышает понимание производственных требований и распределяет их на более обоснованном уровне.

Необходимым условием для осуществления процесса разработки аппаратной части является проведение трассировки и отслеживание требований безопасности. Наихудшее отказное состояние может быть использовано только в случае невозможности проведения трассировки.

Интеграция программного обеспечения и аппаратной части может производиться на компьютерных эмуляторах, опытных образцах или на специальных изделиях, созданных для летных или лабораторных испытаний. Условия определяются в зависимости от вида системы. В результате испытаний получают физическое изделие с гарантийными данными уровня разработки, а так же с настраиваемой конфигурацией.

Одно изделие последовательно интегрируется за другим до полного завершения интеграции системы.

При возникновении трудностей в прогнозировании или моделировании среды, в которой планируется использование ВС, работу могут выполняться непосредственно на ВС. Несмотря на то, что при проведении интеграции на ВС получаю достоверные и экономические эффективные результаты, в условиях модели или лаборатории можно достигнуть впечатляюще высоких результатов.

Гарантии проекта оборудования

Интегрированная система существует для гарантии установки на ВС отдельного и совместного функционирования. Задачей интеграции является доказательство того, что меж-

системные требования изделия будут соответствовать стандартам, а так же будет существовать возможность определить и исправить не предусмотренные функции.

Ключевые моменты, определенные процессом доказательства гарантии:

1. Своевременное определение исполнения поддерживающих процессов;
2. Оценка безопасности, взаимодействующая с другими системами поддержки ключевых точек проекта.

Большинство сложных и высокоинтегрированных систем использует доказательство обеспечения гарантии в течении основной части периода проектирования. Весь процесс сертификации, в данном случае, рассматривается как процесс поддержки проектирования.

Доказательство соответствия ВС, а так же его систем ТНЛГ (требованиям норм летной годности), является основной целью процесса сертификации.

Необходимость обеспечения координации действий заявителя с органом сертификации обусловлена тем, что использование методов гарантии проектирования является главным средством очевидности сертификации интегрированных и сложных систем. Данную процедуру рекомендуется осуществить как можно быстрее (до реализации проекта). Сертификационный орган оценивает полноту данных для доказательства соответствия методов оценки, которые предлагает заявитель.

Аналитическая очевидность соответствия ТНЛГ определяет процесс оценки безопасности. В него входят конкретные оценки, изменяющиеся вместе с проектированием системы и связанные с остальными процессами поддержки ее разработки. Основные процессы определения безопасности[1]:

1. Оценка функциональных рисков (FHA) – производится рассмотрение функций ВС, а так же его систем с целью определения возможных отказов, проведение классификации опасностей и присущих им отказных состояний. Данная оценка выполняется на начальном этапе проектирования и дополняется по мере поступления новых отказных состояний или новых функций.
2. Предварительная оценка безопасности системы (PSSA) – результатом данной процедуры является установление конкретных требований безопасности системы, а так же ее составных изделий. Позднее дается первоначальное подтверждение возможности удовлетворения данных требований архитектурой системы. В процессе проектирования системы предварительная оценка безопасности может корректироваться.
3. Оценка безопасности системы (SSA) – сбор, анализ и документирование доказательств, указывающих на то, что спроектированная система удовлетворяет требованиям безопасности, указанным в раках процесса оценки функциональных опасностей (FHA) и предварительной оценки (PSSA).
4. Анализ общих причин отказов (CCA) – определение и оценка требований по функциональному и физическому разделению систем. В то же время происходит проверка выполнения данных требований.

Далее выполняется процесс валидации, к целям которого относятся проверка правильности и полноты требований.

При верификации могут использоваться отдельные методы валидации. Определение планов верификации и валидации могут обуславливать данное течение процесса.

Проведение верификации дает гарантию на удовлетворение предъявляемых требований к реализуемой системе (если та прошла верификацию).

Выводы

Анализируя зарубежные подходы и процедуры сертификации, валидации и верификации изделий авиационной техники, можно определить, что отличительной чертой зарубеж-

ных технологий проектирования является сертификационная направленность подавляющего большинства видов работ. Работы производятся по принципу сквозной сертификации, для реализации которого на этапе эскиза разрабатывается система сертификаций, охватывающая все виды работ.

Лабораторно стендовые испытания также должны включать в себя процедуры верификации, для придания им сертификационной направленности. Сквозная сертификация должна гарантировать готовность сертификационной документации более чем на 50% к моменту начала летных испытаний.

Сами летные испытания, играющие важную роль для оценки основных характеристик ВС в отечественной практике, за рубежом имеют существенно меньшее влияние на процесс проектирования. Это дает возможность получить до 80% всех данных на земле, что гарантирует собой более высокий экономический эффект от разработки.

Библиографический список

1. Елесина С.И., Никифоров М.Б. Документальное сопровождение разработки и производства радиоэлектронной аппаратуры [Учебное пособие] Рязань, 2014. - 60с.
2. Муратов Е.Р. Методы сокращения вычислительной сложности алгоритмов совмещения разнородных изображений [Статья] Электронный журнал Cloud of Science, Рязань, 2014. - 328-336 с.
3. Новиков А.И., Муратов Е.Р., Логинов А.А., Никифоров М.Б. Сокращение вычислительной сложности совмещения изображений в авиационных системах технического зрения [Статья] Динамика сложных систем, XXI век, №1, т.9, 2015, Москва. - 33-40с.
4. Кузнецов А.Г. «Навигация и управление летательными аппаратами». 2010 –(86) №2.

СОДЕРЖАНИЕ

ИНФОРМАЦИЯ О МЕЖДУНАРОДНОМ ФОРУМЕ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ» СТНО-2018».....	3
МЕЖДУНАРОДНАЯ НАУЧНО-ТЕХНИЧЕСКАЯ КОНФЕРЕНЦИЯ «СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ. ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ».....	5
СЕКЦИЯ «АЛГОРИТМИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ»	5
Цветков И.А. Валентность функций одиночно-пустого выбора.....	5
Цветков И.А. Функции разбивного выбора.....	10
Коваленко В.В., Минасян Р.К. Разработка информационной системы центра подготовки волонтеров.....	15
Кручок Д. Н. Шумоустойчивость систем идентификации диктора.....	19
Ерхов Р.В. Программная реализация веб-приложения статистического анализа данных с расширяемой архитектурой на платформе ASP.net mvc.....	23
Степанов Н.Н. Программная система проверки совпадения текстов программ.....	27
Мартышкин А.И. Визуальная программная система имитационного моделирования вычислительных систем	30
Мартышкин А.И. Анализ многопроцессорной системы с общей многомодульной памятью.....	33
Мартышкин А.И. Натурный эксперимент по исследованию образца реконфигурируемой вычислительной системы для цифровой обработки сигнала на базе ПЛИС.....	36
Мартышкин А.И., Мартенс-Атюшев Д.С. Имитационное моделирование распределенной системы управления с применением функциональных блоков.....	40
Гиголаев А.В, Тярт Н.А., Швечкова О.Г. Модификация стеганографического метода LSB для повышения секретности передачи сообщения.....	43
Решетова В., Софьин Е. Использование технологий «USER DEFINED TARGETS», «USER DEFINED OBJECTS» и «MULTI-TARGETS» в приложениях дополненной реальности.....	48
Епифанцев А.А. Разработка фреймов для описания и выявления ошибочных действий пользователей интернет-приложения обработки количественных числительных естественных языков....	54
Карасёв Д.Н., Кравченко С.В. Проектирование интерактивного путеводителя по университету.....	58

Николаев А.И.

Проверка качества кода при автоматизированном составлении отчетов по
практическим заданиям..... 62

Золотарёв В.В., Овечкин Г.В., Корябкин В.В.

Дивергентное кодирование с алгоритмом около границы Шеннона..... 65

Балабанов Н.Р.

Криптовалюты в современном мире70

Секция «ЭВМ И СИСТЕМЫ» 74**Саблина В.А., Сергеева А.Д.**

Основные подходы к решению задачи SLAM в робототехнике..... 74

Бикташев Р.А., Петкилев А.А., Балаев К.А.

Разработка устройства синхронизации процессов для многопроцессорных систем..... 80

Сычев А.С.

Разработка безэталонного интегрально-мультипликативного показателя качества
цифровых изображений..... 85

Н.Н. Гринченко, Е.С. Геращенко, В.Ю. Потапова

Разработка программных средств обработки электрокардиосигнала с помощью
технологии информационного анализа..... 91

Гургулов С., Никифоров М.Б.

Программный имитатор электронного регулятора частоты вращения
дизель – генератора 18-9ДГ..... 96

Караев С.А., Муратов Е.Р., Никифоров М.Б., Смыкова М.А.

Автоматизация исследования алгоритмов обработки изображений..... 102

Громов А.Ю., Суслина А.А., Левшин Д.А.

Применение технологии BLOCKCHAIN в задачах анализа кадровых ресурсов..... 107

Геращенко Е.С., Никифоров М.Б., Потапова В.Ю.

Использование языка R для статистической обработки метеорологических данных..... 110

Колчаев Д.А., Орлов С.В.

Программная имитация пятен нерезкости с использованием карты глубины..... 115

Никифоров М.В., Тарасов А.С.

Оптимизация алгоритма двойной интерполяции в задачах анализа высотных данных..... 118

Суслина А.А., Кежватова А.Т.

Применение программно-аппаратных средств для анализа и прогнозирования
трат клиентов..... 124

Баранчиков А.И., Нгуен Н.З.

Снижение избыточности данных в реляционных базах данных за счет проверки
ее структуры методом табло..... 128

Морозов Н.

Виртуальная среда моделирования EVE-NG..... 132

Киселев А., Никифоров М.

Оценка экономической эффективности систем визуализации изображений
для летательных аппаратов..... 135

Левшин Д.А., Петровская Т.А.

Применение Javascript библиотеки react в задаче увеличения производительности
web-страницы..... 140

Петровская Т.А., Кежватова А.Т.

Применение метода карты расстояний в задаче проектирования движения
беспилотного наземного устройства..... 143

Барышникова Ю.

Использование методологии Aris для повышения эффективности Управления
благоустройства города Рязани..... 147

Маркина Ю.

Использование информационно-коммуникативных технологий в области
прогнозирования банкротства предприятий..... 151

Королева Е.П.

Обзор современных методов помехоустойчивого кодирования..... 155

Секция «СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ» 160**Войнов А.С., Дубинин В.Н., Сенокосов И.В., Климкина Л.П., Вяткин В.В.**

Функционально-блочное моделирование системы защиты от перегрузок
цифровых подстанций МЭК 61850.....160

Шоназаров П.М., Холов Ф.Т.

Внедрение АСУ в сферу здравоохранения Республики Таджикистан..... 165

Марьясин О.Ю., Огарков А.А.

Идентификация моделей при проектировании инженерных систем зданий..... 171

Барсуков А., Антоненко А.

Приложение для конфигурирования компонентов нейросетевых
преобразователей частотно-временных параметров сигналов..... 176

Скворцов С.В., В.И. Хрюкин

Анализ экспертных оценок проектов на основе коэффициентов ранговой
корреляции.....181

Перепелкин Д.А., Иванчикова М.А., Бышов В.С., Цыганов И.Ю.

Программный модуль для организации сетевого взаимодействия программ
и программных систем в программно-конфигурируемых сетях.....185

Дашкина Л.С.

Сравнение программного обеспечения, реализующего алгоритм кластеризации
K-MEANS и его модификации.....187

Гурьева М.Н.

Алгоритм автоматического определения количества кластеров в задаче
сегментации гиперспектральных снимков..... 190

Сапрыкин А., Сапрыкина Е.

Особенности графического пользовательского интерфейса программы
балансировки нагрузки в сетях центров обработки данных на основе
генетического алгоритма..... 196

Вишневская В.П.

Свободно распространяемые программные средства для создания экспертных
систем..... 199

Мороз В.

Реализация метода градиентного спуска с помощью библиотеки TENSORFLOW
на языке программирования PYTHON..... 203

Фетисова Т.А.

Разработка модели многопоточного генетического алгоритма для графических
процессоров..... 207

Ширенин И.

Взаимодействие с внешними сервисами посредством API..... 211

Бакулева А.В., Бакулева М.А.

Разработка алгоритма поиска ассоциативных правил слабоструктурированных данных..... 215

Миронов В.В.

Построение средства тестирования обучающихся на основе системы автоматизированного проектирования баз знаний..... 218

С.В. Аникеев, В.В. Миронов

Методологические основы построения программного комплекса синтеза проектных решений в задачах внедрения информационных систем расчетов с населением за жилищно-коммунальные услуги..... 224

Братулин А., Киселев А., Никифоров М.

Оценка экономической эффективности систем визуализации изображений для летательных аппаратов..... 228

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ В НАУКЕ И ОБРАЗОВАНИИ

Научное издание

В 10 томах

Том 4

Под общей редакцией О.В. Миловзорова.

Подписано в печать 15.06.18. Формат 60х84 1/8.

Бумага офсетная. Печать офсетная.

Гарнитура «Times New Roman».

Усл. печ. л. 14,5.

Тираж 100 экз. Заказ № 3334.

Рязанский государственный радиотехнический университет,
Редакционно-издательский центр РГРТУ,
390005, г. Рязань, ул. Гагарина, д. 59/1.
Отпечатано в типографии Book Jet,
390005, г. Рязань, ул. Пушкина, д. 18