

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**



ИНФОРМАТИКА И КИБЕРНЕТИКА

4(6)

Донецк – 2016

УДК 004.3+004.9+004.2+51.7+519.6+519.7

**ИНФОРМАТИКА И КИБЕРНЕТИКА, № 4(6), – Донецк: ДонНТУ,
2016.**

Представлены материалы по вопросам приоритетных направлений научно-технического обеспечения в области информатики, кибернетики и вычислительной техники.

Материалы предназначены для специалистов народного хозяйства, ученых, преподавателей, аспирантов и студентов высших учебных заведений.

Редакционная коллегия

Главный редактор: Павлыш В.Н., д.т.н., проф.

Зам. глав. ред.: Андрюхин А.И., к.т.н., с.н.с.

Ответственный секретарь: Завадская Т.В., к.т.н., доц.

Члены редакционной коллегии: Аверин Г.В., д.т.н., проф. Аноприенко А.Я., к.т.н., проф.

Григорьев А.В., к.т.н., доц., Грунский И.С., к.т.н., проф., Зинченко Ю.Е., к.т.н., доц.

Звягинцева А.В., к.т.н., доц., Миненко А.С., д.т.н., проф., Скобцов Ю.А., д.т.н., проф.,

Шелепов В.Ю., д.т.н., проф., Карабчевский В.В., к.т.н., доц. Федяев О.И., к.т.н., доц.

Рекомендовано к печати ученым советом ГВУЗ «Донецкий национальный технический университет» Министерства образования и науки ДНР. Протокол № 9 от 23.12.2016 г.

Журнал зарегистрирован в Министерстве образования и науки ДНР № 307 от 06.08.2015

Контактный адрес редакции

ДонНТУ, ул. Артема, 58, Донецк, 83001

Тел.: +380 (62) 301-08-56. Эл.почта: infcyb.donntu@yandex.ru

Интернет: <http://infcyb.donntu.org>

© Донецкий национальный технический университет
Министерство образования и науки ДНР, 2016

СОДЕРЖАНИЕ

Рефлексивные булевы функции Андрюхин А.И.	4
Анализ UDP-трафика Бельков Д.В., Едемская Е.Н.	14
Разработка информационной системы историй болезни органов брюшной полости пациентов.. Борота А.В., Борота А.А., Кухто А.П., Базиян-Кухто Н.К., Федяев О.И., Бакаленко В.С.	21
Принятие решений в системах с субъективной формой понимания интересов и представлений о ситуации выбора Виноградов Г.П., Виноградова Н.Г.	28
Землянская С.Ю., Сложеницын А.В., Андриевская Н.К. Структура модели для прогнозирования поведения групп лиц.	37
Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС Android». Коломойцева И.А., Завадская Т.В.	44
База данных учета кадров организации Костюк А.И., Лебедева Е.О.	52
Программная реализация алгоритмов обфускации программного кода языка JavaScript Медгаус С.В., Чернышова А.В.	59
Моделирование и информационные технологии при распознании лица человека по его мимическим фотографиями. Миненко А.С.	67
Механизм срабатывания модулей знаний интеллектуальной оболочки.. Морозова О.В.	73
Исследование использования дельта-преобразований для определения местоположения беспилотного летательного аппарата Пирская Л.В., Кравченко П.П.	81
Формальная верификация циклических программ. Федяев О.И.	89
Компьютерное исследование и визуализация колебательной динамики систем в комплексной форме Филер С.Е., Андрюхин А.И.	98
Электронное обучение (e-Learning).. Черкасов В.А., Азаров В.Н.	109
Автоматическая генерация структуры Wiki-систем при помощи онтологий. Шестаков В.К.	116

УДК 004.0- 517.8- 621.3

Рефлексивные булевы функции

А.И. Андрияхин

Донецкий национальный технический университет, г. Донецк,
alexandruckin@rambler.ru

Андрияхин А.И. Рефлексивные булевы функции. Эта работа относится к фундаментальной проблеме определения базовых примитивов самосознания. Человеческий мозг основываясь на их свойствах, способен самоорганизовываться и адаптироваться во внешней среде. В работе определены классы рефлексивных булевых функций. Выполнена их классификация. Все рефлексивные булевы функции от двух и трех переменных приведены. Также все рефлексивные булевы функции от четырех переменных определены. Показано, что логическая операция импликация принадлежит к классу рефлексивных функций. Результаты компьютерных экспериментов приведены. Программная система Вольфрам Математика использовалась в расчетах.

Ключевые слова: рефлексия, булевы функции, самоорганизация, самосознание, мозг

Введение

Рефлексия – принцип человеческого мышления, направляющий его на осмысление и осознание собственных форм и предпосылок; способствует формированию самосознания, «оборачивая» сознание само на себя. Многие учёные, философы считают, что рефлексия отличает человека от животного, и рассматривают её как универсальный способ анализа самосознания. Так, Тейяр де Шарден усматривал основное различие между человеком и животным в степени развития рефлексии [1]. Представляя самосознание аспектом сознания, Выготский подчеркивал значение рефлексивности как конституирующей характеристики сознания, указывая, что последнее возникает лишь с появлением самосознания. При рассмотрении в своих ранних работах механизмов сознания, Выготский сближал сознание и самосознание.

Важный момент в развитии рефлексии представляет появление вербального отражения собственных процессов и действий, выступающего, по Выготскому, основой для развития самосознания и высших волевых регулирующих механизмов. Он указывал, что переход к словесной интроспекции повторяет в общих чертах аналогичное развитие восприятия внешнего мира – переход от «бессловесного и, следовательно, несмыслового восприятия» к восприятию смысловому, словесному, предметному, т. е. обобщенному. Поэтому и переход к словесной интроспекции не означает ничего другого, кроме начинающегося обобщения внутренних форм активности.

Развитие речевой функции в процессе гоминизации открыло перед центральной нервной системой совершенно новые возможности для формирования рефлексивных, моделирующих и управляющих механизмов. Может быть, самой

принципиальной из них оказалась возможность сколь угодно точной рефлексии сколь угодно высокого уровня, достижимой лишь на уровне социума, а не индивидуума [2-4].

Самосознание играет особую роль в процессе получения знания – это знание особого рода. Сократ считал, что главное – это самосознание (духовное самосовершенствование). Согласно взглядам Платона и Аристотеля, самосознание – это атрибут Бога, единство мыслимого и мысли. Выготский обозначал термином «самосознание» следующий этап развития рефлексивных процессов, основанный на понятийном мышлении. Он выделял самосознание как особый, достаточно поздний этап в развитии сознания и личности. Оно понималось им, как более высокий психический синтез, проявляющийся в образовании новых связей между различными функциями – третичных высших функций. Новизна их обусловлена тем, что в их основе лежит рефлексия, отражение собственных процессов в сознании. Это ведёт к перестройке всей структуры сознания человека: его личность начинает участвовать в каждом отдельном акте, опосредовать связи функций между собой. Согласно Выготскому, в то время как психологические системы (включающие внимание, память, мышление и другие функции) строятся на вторичных (по отношению к определяемым конституцией организма) биологических связях, личность (с ее самосознанием, рефлексией, самооценкой) – на особых, третичных связях. Существование этих третичных связей проявляется в том функциональном значении, которое отдельные функции имеют в общей структуре личности (например, сновидение в поведении примитивного человека выполняет ту же

функцию, которую в нашем поведении выполняет мышление).

Философский аспект анализа самосознания связан с выявлением его гносеологической сущности, выяснением его соотношения с объективным бытием личности. Осознавая существование мира объектов, того, что находится вне нас, мы отделяем себя от внешнего мира. Там – объект, а здесь – Я. В процессе осознания объекта всегда присутствует скрытая черта – не-Я. Возникает полярное взаимоотношение объекта (не-Я) и субъекта (Я). Принадлежность образов объектов именно моему Я вызывает у человека представление о себе как об особой реальности, противостоящей миру объектов и вместе с тем, отличной от других подобных ему Я.

Выделив свое Я, мы можем смотреть на себя как на нечто самостоятельное по отношению к себе же, т. е. как на объект. Способность к самоотражению своего Я заключает в себе и одно из важнейших оснований объективного исследования самосознания. В том, как личность представляет свое Я, отражается мера ее осознания себя и уровень зрелости личности в целом. Вместе с тем в способности взглянуть на себя, как на нечто иное, и коренятся трудности исследования самосознания, так как «вынесение» личностью себя «вовне» связано с целым рядом субъективных ее особенностей, часто препятствующих созданию объективного представления о своем Я.

При детальном рассмотрении рефлексии и самосознания проявляются их необычные свойства. Так, многие философы указывали, что познающий субъект не может быть объектом собственного опыта, т. е. объектом собственного знания. Знание, опыт направлены на мир внешних объектов, и мы можем знать состояния и отношения физических предметов. Если я познаю объект, то могу ли я одновременно познавать и самого себя, познающего и акт собственного познания? Не приводит ли это к логическому парадоксу, когда высказывание в качестве одного из референтов имеет самого себя? Так, например, мои глаза принципиально не могут видеть самих себя и процесс собственного видения; линзы перископа отображают окружающее, но не могут отобразить самих себя[5].

Фреге в известной работе указал на непринадлежность мысли ни объектам внешнего мира, ни представлениям внутреннего мира человека[6]. Следовательно, если мышление есть формулирование мысли (суждение-констатация истинности мысли и его выражение-утверждение), отсюда следует, что можно соотносить рефлексии физический процесс, производный от другого базового, в котором основными значениями сигналов являются двоичные. На эту роль в первую очередь можно

отнести волны – как элементы, которые в основном переносят энергию, а не вещество.

Человеческая способность представлять мысленно собственные мысли и чувства (т.е.рефлексия) была центральной темой науки, начиная с Джона Локка.

Согласно работам Лефевра, субъект, обладающим рефлексивным сознанием может быть изображен, как миниатюрная человеческая фигура с образ себя внутри его головы. Он же подчеркивает, что судя по всему человеку от рождения даны фундаментальные рефлексивные структуры с двумя рангами рефлексии (субъект чувствует себя и чувствует себя чувствующим себя) и автоматическим механизмом счета.

На наш взгляд, проблема адекватного отражения рефлексии является одной из центральных в ИИ. Однако большинство исследователей ИИ не уделяют ей должного внимания, сосредотачиваясь на попытках воссоздания лишь отдельных процессов, моделирующих различные функции мышления человека. На сегодняшний день можно выделить следующие подходы к этой проблеме, представленные на рис.1[7].



Рисунок 1 – Отражение рефлексии

Первый метод развивается в русле автоэпистемологических рассуждений [8], основой которых является построение дерева альтернативных возможностей, полный перебор вариантов которого или поиск с известными правилами его направленности приводит к требуемому результату. Рассмотрение вопросов, связанных с реализацией системы самодиагностики на дискретном уровне является предметом работы [9]. Сразу отметим характерные черты данного подхода. Т. к. данный

подход полностью формален (логический вывод осуществляется на системах булевых уравнений), то ему присущи и типичные ограничения формальных систем. Булева логика часто не позволяет адекватно охарактеризовать объект, так что её применение приводит к разного рода парадоксам, аналогичным парадоксам Рашара, Греллинга, Рассела. Основной причиной этого является т. н. проблема самоприменимости или самоотносимости машин Тьюринга. Т. е. невозможно средствами некоторой формальной системы исследовать её саму или иначе, некая сущность, объект, множество, система не может быть охарактеризована через совокупность, которой она принадлежит.

Второй подход к проблеме связан с отражением рефлексии в непрерывных средах, в частности, в осцилляторной нейронной сети. Отметим следующее: предлагается рассматривать осциллятор, построенный из множества связанных нейронов, как некий «элемент сознания». В этом случае рефлексия можно интерпретировать в терминах колебаний сети связанных осцилляторов как, например, взаимодействие этих осцилляторов в присутствии шума[10].

Данная работа примыкает к известной математико-психологической модели Лефевра, использующей функцию $X1 = F(x1, F(x2, x3))$ [11].

Функция $X2 = F(x2, x3)$ интерпретируется как образ себя, имеющийся у субъекта. Первая переменная этой функции, представляет перцептивный вход, а второй переменной соответствует ментальный образ себя.

Получен известный результат, что функциональное уравнение

$$\Phi(x1, \Phi(x2, x3)) = x1 + (1 - x1)(1 - x2)x3$$

где $x1, x2, x3$ числа из $[0,1]$ и все значения $\Phi(x, y)$ принадлежат $[0,1]$, имеет единственное решение

$$\Phi(x, y) = 1 - y + xy = F(x, y).$$

По аналогии с этим результатом рассмотрим рефлексивные булевы функции i -ого порядка, которые обладают свойством

$$\Phi(x_1, x_2, \dots, x_{i-1}, \Phi(x_1, x_2, \dots, x_n), x_{i+1}, \dots, x_n) = \Phi(x_1, x_2, \dots, x_n).$$

Обозначим через $\Phi_{i,j}$ множество рефлексивных булевых функций i -порядка от j переменных. Также обозначим через Φ_j множество рефлексивных булевых функций от j переменных.

Решаемая в статье задача состоит в расширении и построении рефлексивных булевых функций зависящих от 2, 3, 4 и более переменных.

Анализ функции $\Phi(x, y) = 1 - Y + XY$

Существует известное правило перехода от булевых функций с двоичными переменными к непрерывным функциям от действительных

переменных, для которых значение функции и переменных находятся в интервале $(0,1)$.

Для этого мы выполняем замену булевых выражений алгебраическими выражениями, согласно $X \vee Y = X + Y - XY$, $X \wedge Y = XY$, $\bar{X} = 1 - X$, $X \oplus Y = X + Y - 2XY$.

Здесь X, Y в правых частях равенств являются положительными действительными переменными не больше 1.

Сопоставим действительным переменным X, Y булевы переменные B и A и получим формулу $\Phi(B, A) = 1 - A + AB$.

Построим таблицу истинности для $\Phi(B, A)$ и операции импликации, которую представим в таб.1.

Табл.1.–Таблица истинности $\Phi(B, A)$ и импликации

	A	B	$\Phi(B, A)$	$A \rightarrow B$
1	F	F	T	T
2	F	T	T	T
3	T	F	F	F
4	T	T	T	T

Следовательно, эта функция представляет собой вещественное расширение логической операции импликации с переменной порядка аргументов.

Рефлексивность мышления и правила МР

Выполним анализ в свете вышеизложенного таких важных операций, как импликацию и правило логического вывода «modus ponens» (MP).

Рассмотрим любые два высказывания A, B . Если составить из этих двух высказываний A, B сложное высказывание, если A , то B , (обозначается как $A \rightarrow B$), то мы определяем логическую операцию импликации и ее таблица истинности имеет вид согласно таб.1.

Заметим, что первые две строчки таблицы (A ложно) иногда интерпретируют как "Из ложного высказывания следует что угодно", что вызывает у части специалистов резкие возражения. Совокупность 2 и 4 строчки таблицы (B истинно) часто формулируют, как "Истинное высказывание следует из чего угодно".

Анализ выражения $A \rightarrow B$ приводит к заключению, что в правой части опускают посылку B , т.е. импликация является логической связкой с обратной связью и может быть представлена рис.2. Более того, существование релевантной логики базируется на требовании наличия общих переменных по обе стороны правила логического вывода и следовательно любой логический вывод есть правило с обратной связью.

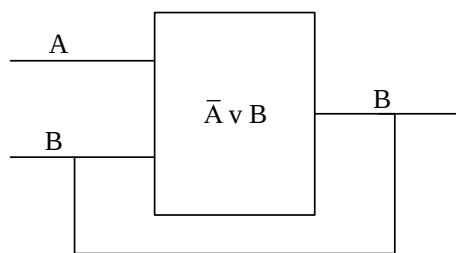


Рисунок 2 – Рефлексивная модель импликации.

Действительно, $\neg A \vee (\neg A \vee B) = \neg A \vee \neg A \vee B = \neg A \vee B$. Аналогичные рассуждения для правила МР приводят к рис.3.

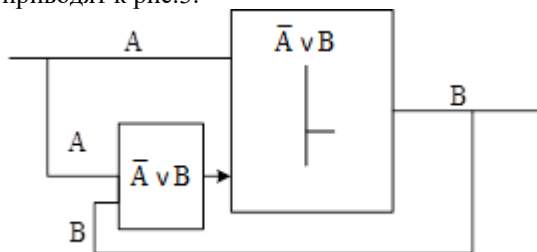


Рисунок 3 – Рефлексивная модель МР.

Следуя этим рисункам можно построить аппаратную реализацию этих схем [12-14].

Рефлексивные булевы функции от двух переменных

Общее число всех булевых функций от n переменных определяется формулой $2^{f(n)}$, где $f(n)=2^n$. Отсюда следует известный результат, что число булевых функций от 2 переменных равно 16, число булевых функций от 3 переменных равно 256, число булевых функций от 4 переменных равно 65536 и т.д.

Для функций от двух переменных мы имеем рефлексивные формы 1 и 2 порядка, представленные на рис.3.

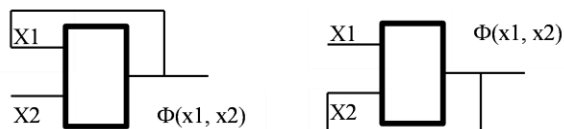


Рисунок 3 – Рефлексивные булевы функции 1 и 2 порядка.

Ниже представлен список из 4 (из общего числа 9) нетривиальных булевых выражений 1-порядка от двух переменных, для которых выполняется соотношение $\Phi(x1, x2) = \Phi(\Phi(x1, x2), x2)$ и в выражении для этих функций присутствуют две переменные $x1, x2$.

- | | | |
|---|---|---------------------|
| 1 | 4 | $x1 \wedge \neg x2$ |
| 2 | 8 | $x1 \wedge x2$ |

- | | | |
|---|----|-------------------|
| 3 | 13 | $x1 \vee \neg x2$ |
| 4 | 14 | $x1 \vee x2$ |

Аналогично мы можем представить список из 4 (из общего числа 9) нетривиальных булевых выражений 2-порядка от двух переменных, для которых выполняется соотношение $\Phi(x1, x2) = \Phi(x1, \Phi(x1, x2))$ и в выражении для этих функций присутствуют две переменные $x1, x2$.

- | | | |
|---|----|---------------------|
| 1 | 2 | $\neg x1 \wedge x2$ |
| 2 | 8 | $x1 \wedge x2$ |
| 3 | 11 | $\neg x1 \vee x2$ |
| 4 | 14 | $x1 \vee x2$ |

Второе число в строке описания этих функций и далее в других списках означает их номер в системе Wolfram Mathematica (оператор BooleanFunction).

Пересечением множеств $\Phi_{1,2}$ и $\Phi_{2,2}$ являются функции $x1 \vee x2$ и $x1 \wedge x2$.

Рефлексивные булевы функции первого порядка от трех переменных

Начиная с этого раздела будем использовать вместо знака отрицания $\neg$ символ !. На рис.4 представлена рефлексивная булева функция 1 порядка, так как

$$\Phi(x1, x2, x3) = \Phi(\Phi(x1, x2, x3), x2, x3).$$

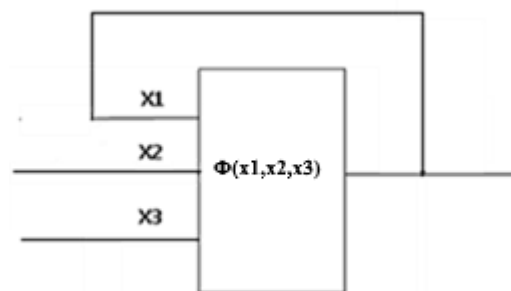


Рисунок 4. –Рефлексивная булева функция от трех переменных 1 порядка

Ниже представлен список из 56 нетривиальных булевых выражений от трех переменных 1 порядка.

- | | | |
|----|-----|--|
| 1 | 16 | $x1 \wedge !x2 \wedge !x3$ |
| 2 | 32 | $x1 \wedge !x2 \wedge x3$ |
| 3 | 49 | $(x1 \wedge !x2) \vee (!x2 \wedge !x3)$ |
| 4 | 50 | $(x1 \wedge !x2) \vee (!x2 \wedge x3)$ |
| 5 | 64 | $x1 \wedge x2 \wedge !x3$ |
| 6 | 81 | $(x1 \wedge !x3) \vee (!x2 \wedge !x3)$ |
| 7 | 84 | $(x1 \wedge !x3) \vee (x2 \wedge !x3)$ |
| 8 | 96 | $(x1 \wedge x2 \wedge !x3) \vee (x1 \wedge !x2 \wedge x3)$ |
| 9 | 98 | $(x1 \wedge x2 \wedge !x3) \vee (!x2 \wedge x3)$ |
| 10 | 100 | $(x1 \wedge !x2 \wedge x3) \vee (x2 \wedge !x3)$ |
| 11 | 112 | $(x1 \wedge !x2) \vee (x1 \wedge !x3)$ |
| 12 | 113 | $(x1 \wedge !x2) \vee (x1 \wedge !x3) \vee (!x2 \wedge !x3)$ |
| 13 | 114 | $(x1 \wedge !x2) \vee (x1 \wedge !x3) \vee (!x2 \wedge x3)$ |
| 14 | 115 | $(x1 \wedge !x3) \vee !x2$ |
| 15 | 116 | $(x1 \wedge !x2) \vee (x2 \wedge !x3)$ |

16 117 $(x1 \wedge !x2) \vee !x3$
 17 118 $(x1 \wedge !x2) \vee (x2 \wedge !x3) \vee (!x2 \wedge x3)$
 18 128 $x1 \wedge x2 \wedge x3$
 19 144 $(x1 \wedge x2 \wedge x3) \vee (x1 \wedge !x2 \wedge !x3)$
 20 145 $(x1 \wedge x2 \wedge x3) \vee (!x2 \wedge !x3)$
 21 152 $(x1 \wedge !x2 \wedge !x3) \vee (x2 \wedge x3)$
 22 162 $(x1 \wedge x3) \vee (!x2 \wedge x3)$
 23 168 $(x1 \wedge x3) \vee (x2 \wedge x3)$
 24 176 $(x1 \wedge !x2) \vee (x1 \wedge x3)$
 25 177 $(x1 \wedge !x2) \vee (x1 \wedge x3) \vee (!x2 \wedge !x3)$
 26 178 $(x1 \wedge !x2) \vee (x1 \wedge x3) \vee (!x2 \wedge x3)$
 27 179 $(x1 \wedge x3) \vee !x2$
 28 184 $(x1 \wedge !x2) \vee (x2 \wedge x3)$
 29 185 $(x1 \wedge !x2) \vee (x2 \wedge x3) \vee (!x2 \wedge !x3)$
 30 186 $(x1 \wedge !x2) \vee x3$
 31 196 $(x1 \wedge x2) \vee (x2 \wedge !x3)$
 32 200 $(x1 \wedge x2) \vee (x2 \wedge x3)$
 33 208 $(x1 \wedge x2) \vee (x1 \wedge !x3)$
 34 209 $(x1 \wedge x2) \vee (!x2 \wedge !x3)$
 35 212 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (x2 \wedge !x3)$
 36 213 $(x1 \wedge x2) \vee !x3$
 37 216 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (x2 \wedge x3)$
 38 217 $(x1 \wedge x2) \vee (x2 \wedge x3) \vee (!x2 \wedge !x3)$
 39 220 $(x1 \wedge !x3) \vee x2$
 40 224 $(x1 \wedge x2) \vee (x1 \wedge x3)$
 41 226 $(x1 \wedge x2) \vee (!x2 \wedge x3)$
 42 228 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge !x3)$
 43 230 $(x1 \wedge x2) \vee (x2 \wedge !x3) \vee (!x2 \wedge x3)$
 44 232 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge x3)$
 45 234 $(x1 \wedge x2) \vee x3$
 46 236 $(x1 \wedge x3) \vee x2$
 47 241 $x1 \vee (!x2 \wedge !x3)$
 48 242 $x1 \vee (!x2 \wedge x3)$
 49 244 $x1 \vee (x2 \wedge !x3)$
 50 246 $x1 \vee (x2 \wedge !x3) \vee (!x2 \wedge x3)$
 51 247 $x1 \vee !x2 \vee !x3$
 52 248 $x1 \vee (x2 \wedge x3)$
 53 249 $x1 \vee (x2 \wedge x3) \vee (!x2 \wedge !x3)$
 54 251 $x1 \vee !x2 \vee x3$
 55 253 $x1 \vee x2 \vee !x3$
 56 254 $x1 \vee x2 \vee x3$

Рефлексивные булевы функции второго порядка от трех переменных

На рис.5 представлена рефлексивная булева функция 2 порядка, так как $\Phi(x1, x2, x3) = \Phi(x1, \Phi(x1, x2, x3), x3)$.

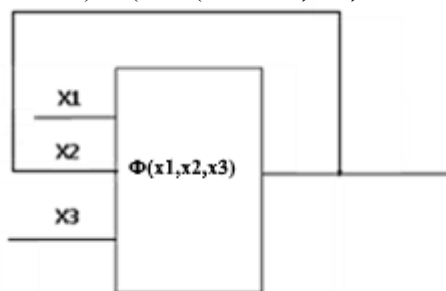


Рисунок 5. –Рефлексивная булева функция от трех переменных 2 порядка

Ниже представлен список из 54 нетривиальных булевых выражений от трех переменных 2 порядка

1 4 $!x1 \wedge x2 \wedge !x3$
 2 8 $!x1 \wedge x2 \wedge x3$
 3 13 $(!x1 \wedge x2) \vee (!x1 \wedge !x3)$
 4 14 $(!x1 \wedge x2) \vee (!x1 \wedge x3)$
 5 64 $x1 \wedge x2 \wedge !x3$
 6 69 $(!x1 \wedge !x3) \vee (x2 \wedge !x3)$
 7 72 $(x1 \wedge x2 \wedge !x3) \vee (!x1 \wedge x2 \wedge x3)$
 8 74 $(x1 \wedge x2 \wedge !x3) \vee (!x1 \wedge x3)$
 9 76 $(!x1 \wedge x2) \vee (x2 \wedge !x3)$
 10 77 $(!x1 \wedge x2) \vee (!x1 \wedge !x3) \vee (x2 \wedge !x3)$
 11 78 $(!x1 \wedge x2) \vee (!x1 \wedge x3) \vee (x2 \wedge !x3)$
 12 79 $!x1 \vee (x2 \wedge !x3)$
 13 84 $(x1 \wedge !x3) \vee (x2 \wedge !x3)$
 14 88 $(x1 \wedge !x3) \vee (!x1 \wedge x2 \wedge x3)$
 15 92 $(x1 \wedge !x3) \vee (!x1 \wedge x2)$
 16 93 $(!x1 \wedge x2) \vee !x3$
 17 128 $x1 \wedge x2 \wedge x3$
 18 132 $(x1 \wedge x2 \wedge x3) \vee (!x1 \wedge x2 \wedge !x3)$
 19 133 $(x1 \wedge x2 \wedge x3) \vee (!x1 \wedge !x3)$
 20 138 $(!x1 \wedge x3) \vee (x2 \wedge x3)$
 21 140 $(!x1 \wedge x2) \vee (x2 \wedge x3)$
 22 141 $(!x1 \wedge x2) \vee (!x1 \wedge !x3) \vee (x2 \wedge x3)$
 23 142 $(!x1 \wedge x2) \vee (!x1 \wedge x3) \vee (x2 \wedge x3)$
 24 143 $!x1 \vee (x2 \wedge x3)$
 25 164 $(x1 \wedge x3) \vee (!x1 \wedge x2 \wedge !x3)$
 26 168 $(x1 \wedge x3) \vee (x2 \wedge x3)$
 27 172 $(x1 \wedge x3) \vee (!x1 \wedge x2)$
 28 174 $(!x1 \wedge x2) \vee x3$
 29 196 $(x1 \wedge x2) \vee (x2 \wedge !x3)$
 30 197 $(x1 \wedge x2) \vee (!x1 \wedge !x3)$
 31 200 $(x1 \wedge x2) \vee (x2 \wedge x3)$
 32 202 $(x1 \wedge x2) \vee (!x1 \wedge x3)$
 33 205 $(!x1 \wedge !x3) \vee x2$
 34 206 $(!x1 \wedge x3) \vee x2$
 35 208 $(x1 \wedge x2) \vee (x1 \wedge !x3)$
 36 212 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (x2 \wedge !x3)$
 37 213 $(x1 \wedge x2) \vee !x3$
 38 216 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (x2 \wedge x3)$
 39 218 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (!x1 \wedge x3)$
 40 220 $(x1 \wedge !x3) \vee x2$
 41 222 $(x1 \wedge !x3) \vee (!x1 \wedge x3) \vee x2$
 42 223 $!x1 \vee x2 \vee !x3$
 43 224 $(x1 \wedge x2) \vee (x1 \wedge x3)$
 44 228 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge !x3)$
 45 229 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (!x1 \wedge !x3)$
 46 232 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge x3)$
 47 234 $(x1 \wedge x2) \vee x3$
 48 236 $(x1 \wedge x3) \vee x2$
 49 237 $(x1 \wedge x3) \vee (!x1 \wedge !x3) \vee x2$
 50 239 $!x1 \vee x2 \vee x3$
 51 244 $x1 \vee (x2 \wedge !x3)$
 52 248 $x1 \vee (x2 \wedge x3)$
 53 253 $x1 \vee x2 \vee !x3$
 54 254 $x1 \vee x2 \vee x3$

**Рефлексивные булевы функции
третьего порядка от трех переменных**

Число всех рефлексивных булевых функций равно 81, так как помимо нетривиальных к ним добавляются False, True, x_1 и некоторые другие.

На рис.6 представлена рефлексивная булева функция 3 порядка, так как $\Phi(x_1, x_2, x_3) = \Phi(x_1, x_2, \Phi(x_1, x_2, x_3))$.

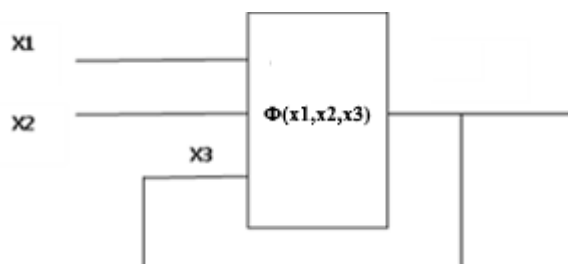


Рисунок 6. –Рефлексивная булева функция от трех переменных 3 порядка.

Ниже представлен список из 56 нетривиальных булевых выражений от трех переменных, которые назовем рефлексивными булевыми функциями 3 порядка.

Мы считаем их нетривиальными, так как в выражении для этих функций присутствуют все три переменные x_1, x_2, x_3 .

- 1 2 $\neg x_1 \wedge \neg x_2 \wedge x_3$
- 2 8 $\neg x_1 \wedge x_2 \wedge x_3$
- 3 11 $(\neg x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_3)$
- 4 14 $(\neg x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3)$
- 5 32 $x_1 \wedge \neg x_2 \wedge x_3$
- 6 35 $(\neg x_1 \wedge \neg x_2) \vee (\neg x_2 \wedge x_3)$
- 7 40 $(x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2 \wedge x_3)$
- 8 42 $(\neg x_1 \wedge x_3) \vee (\neg x_2 \wedge x_3)$
- 9 43 $(\neg x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_3) \vee (\neg x_2 \wedge x_3)$
- 10 44 $(x_1 \wedge \neg x_2 \wedge x_3) \vee (\neg x_1 \wedge x_2)$
- 11 46 $(\neg x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3)$
- 12 47 $\neg x_1 \vee (\neg x_2 \wedge x_3)$
- 13 50 $(x_1 \wedge \neg x_2) \vee (\neg x_2 \wedge x_3)$
- 14 56 $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2 \wedge x_3)$
- 15 58 $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_3)$
- 16 59 $(\neg x_1 \wedge x_3) \vee \neg x_2$
- 17 62 $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3)$
- 18 128 $x_1 \wedge x_2 \wedge x_3$
- 19 130 $(x_1 \wedge x_2 \wedge x_3) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3)$
- 20 131 $(x_1 \wedge x_2 \wedge x_3) \vee (\neg x_1 \wedge \neg x_2)$
- 21 138 $(\neg x_1 \wedge x_3) \vee (x_2 \wedge x_3)$
- 22 139 $(\neg x_1 \wedge \neg x_2) \vee (x_2 \wedge x_3)$
- 23 140 $(\neg x_1 \wedge x_2) \vee (x_2 \wedge x_3)$
- 24 142 $(\neg x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3) \vee (x_2 \wedge x_3)$
- 25 143 $\neg x_1 \vee (x_2 \wedge x_3)$
- 26 162 $(x_1 \wedge x_3) \vee (\neg x_2 \wedge x_3)$
- 27 163 $(x_1 \wedge x_3) \vee (\neg x_1 \wedge \neg x_2)$
- 28 168 $(x_1 \wedge x_3) \vee (x_2 \wedge x_3)$

- 29 171 $(\neg x_1 \wedge \neg x_2) \vee x_3$
- 30 172 $(x_1 \wedge x_3) \vee (\neg x_1 \wedge x_2)$
- 31 174 $(\neg x_1 \wedge \neg x_2) \vee x_3$
- 32 176 $(x_1 \wedge \neg x_2) \vee (x_1 \wedge x_3)$
- 33 178 $(x_1 \wedge \neg x_2) \vee (x_1 \wedge x_3) \vee (\neg x_2 \wedge x_3)$
- 34 179 $(x_1 \wedge x_3) \vee \neg x_2$
- 35 184 $(x_1 \wedge \neg x_2) \vee (x_2 \wedge x_3)$
- 36 186 $(x_1 \wedge \neg x_2) \vee x_3$
- 37 188 $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee (x_2 \wedge x_3)$
- 38 190 $(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2) \vee x_3$
- 39 191 $\neg x_1 \vee \neg x_2 \vee x_3$
- 40 194 $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2 \wedge x_3)$
- 41 200 $(x_1 \wedge x_2) \vee (x_2 \wedge x_3)$
- 42 202 $(x_1 \wedge x_2) \vee (\neg x_1 \wedge x_3)$
- 43 203 $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2) \vee (x_2 \wedge x_3)$
- 44 206 $(\neg x_1 \wedge x_3) \vee x_2$
- 45 224 $(x_1 \wedge x_2) \vee (x_1 \wedge x_3)$
- 46 226 $(x_1 \wedge x_2) \vee (\neg x_2 \wedge x_3)$
- 47 227 $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2) \vee (\neg x_2 \wedge x_3)$
- 48 232 $(x_1 \wedge x_2) \vee (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$
- 49 234 $(x_1 \wedge x_2) \vee x_3$
- 50 235 $(x_1 \wedge x_2) \vee (\neg x_1 \wedge \neg x_2) \vee x_3$
- 51 236 $(x_1 \wedge x_3) \vee x_2$
- 52 239 $\neg x_1 \vee x_2 \vee x_3$
- 53 242 $x_1 \vee (\neg x_2 \wedge x_3)$
- 54 248 $x_1 \vee (x_2 \wedge x_3)$
- 55 251 $x_1 \vee \neg x_2 \vee x_3$
- 56 254 $x_1 \vee x_2 \vee x_3$

**Приближения рефлексивных
булевых функций**

При расчете выходных частот для логических схем мы выполняем замену булевых выражений алгебраическими выражениями, согласно $X \vee Y = X + Y - XY$, $X \wedge Y = XY$, $\bar{X} = 1 - X$, $X \oplus Y = X + Y - 2XY$, где X, Y в правых частях равенств являются положительными действительными переменными не больше 1.

Мы интерпретируем их как частоту появления 1 в двоичной случайной последовательности.

Ниже представлены действительные расширения функций третьего порядка

- 1 2 $(1-x_1) * (1-x_2) * x_3$
- 2 8 $(1-x_1) * x_2 * x_3$
- 3 11 $((1-x_1) * (1-x_2)) + ((1-x_1) * x_3) - ((1-x_1) * (1-x_2)) * ((1-x_1) * x_3)$
- 4 14 $((1-x_1) * x_2) + ((1-x_1) * x_3) - ((1-x_1) * x_2) * ((1-x_1) * x_3)$
- 5 32 $x_1 * (1-x_2) * x_3$
- 6 35 $((1-x_1) * (1-x_2)) + ((1-x_2) * x_3) - ((1-x_1) * (1-x_2)) * ((1-x_2) * x_3)$
- 7 40 $(x_1 * (1-x_2) * x_3) + ((1-x_1) * x_2 * x_3) - (x_1 * (1-x_2) * x_3) * ((1-x_1) * x_2 * x_3)$
- 8 42 $((1-x_1) * x_3) + ((1-x_2) * x_3) - ((1-x_1) * x_3) * ((1-x_2) * x_3)$

9 43 $((1-x_1) * (1-x_2)) + ((1-x_1) * x_3) - ((1-x_1) * (1-x_2)) * ((1-x_1) * x_3) + ((1-x_2) * x_3) - ((1-x_2) * x_3) * ((1-x_1) * (1-x_2)) + ((1-x_1) * x_3) - ((1-x_1) * (1-x_2)) * ((1-x_1) * x_3)$

10 44 $(x_1 * (1-x_2) * x_3) + ((1-x_1) * x_2) - (x_1 * (1-x_2) * x_3) * ((1-x_1) * x_2)$

11 46 $((1-x_1) * x_2) + ((1-x_2) * x_3) - ((1-x_1) * x_2) * ((1-x_2) * x_3)$

12 47 $(1-x_1) + ((1-x_2) * x_3) - (1-x_1) * ((1-x_2) * x_3)$

13 50 $(x_1 * (1-x_2)) + ((1-x_2) * x_3) - (x_1 * (1-x_2)) * ((1-x_2) * x_3)$

14 56 $(x_1 * (1-x_2)) + ((1-x_1) * x_2 * x_3) - (x_1 * (1-x_2)) * ((1-x_1) * x_2 * x_3)$

15 58 $(x_1 * (1-x_2)) + ((1-x_1) * x_3) - (x_1 * (1-x_2)) * ((1-x_1) * x_3)$

16 59 $((1-x_1) * x_3) + (1-x_2) - ((1-x_1) * x_3) * (1-x_2)$

17 62 $(x_1 * (1-x_2)) + ((1-x_1) * x_2) - (x_1 * (1-x_2)) * ((1-x_1) * x_2) + ((1-x_2) * x_3) - ((1-x_2) * x_3) * (x_1 * (1-x_2)) + ((1-x_1) * x_2) - (x_1 * (1-x_2)) * ((1-x_1) * x_2)$

18 128 $x_1 * x_2 * x_3$

19 130 $(x_1 * x_2 * x_3) + ((1-x_1) * (1-x_2) * x_3) - (x_1 * x_2 * x_3) * ((1-x_1) * (1-x_2) * x_3)$

20 131 $(x_1 * x_2 * x_3) + ((1-x_1) * (1-x_2)) - (x_1 * x_2 * x_3) * ((1-x_1) * (1-x_2))$

21 138 $((1-x_1) * x_3) + (x_2 * x_3) - ((1-x_1) * x_3) * (x_2 * x_3)$

22 139 $((1-x_1) * (1-x_2)) + (x_2 * x_3) - ((1-x_1) * (1-x_2)) * (x_2 * x_3)$

23 140 $((1-x_1) * x_2) + (x_2 * x_3) - ((1-x_1) * x_2) * (x_2 * x_3)$

24 142 $((1-x_1) * x_2) + ((1-x_1) * x_3) - ((1-x_1) * x_2) * ((1-x_1) * x_3) + (x_2 * x_3) - (x_2 * x_3) * ((1-x_1) * x_2) + ((1-x_1) * x_3) - ((1-x_1) * x_2) * ((1-x_1) * x_3)$

25 143 $(1-x_1) + (x_2 * x_3) - (1-x_1) * (x_2 * x_3)$

26 162 $(x_1 * x_3) + ((1-x_2) * x_3) - (x_1 * x_3) * ((1-x_2) * x_3)$

27 163 $(x_1 * x_3) + ((1-x_1) * (1-x_2)) - (x_1 * x_3) * ((1-x_1) * (1-x_2))$

28 168 $(x_1 * x_3) + (x_2 * x_3) - (x_1 * x_3) * (x_2 * x_3)$

29 171 $((1-x_1) * (1-x_2)) + x_3 - ((1-x_1) * (1-x_2)) * x_3$

30 172 $(x_1 * x_3) + ((1-x_1) * x_2) - (x_1 * x_3) * ((1-x_1) * x_2)$

31 174 $((1-x_1) * x_2) + x_3 - ((1-x_1) * x_2) * x_3$

32 176 $(x_1 * (1-x_2)) + (x_1 * x_3) - (x_1 * (1-x_2)) * (x_1 * x_3)$

33 178 $(x_1 * (1-x_2)) + (x_1 * x_3) - (x_1 * (1-x_2)) * (x_1 * x_3) + ((1-x_2) * x_3) - ((1-x_2) * x_3) * (x_1 * (1-x_2)) + (x_1 * x_3) - (x_1 * (1-x_2)) * (x_1 * x_3)$

34 179 $(x_1 * x_3) + (1-x_2) - (x_1 * x_3) * (1-x_2)$

35 184 $(x_1 * (1-x_2)) + (x_2 * x_3) - (x_1 * (1-x_2)) * (x_2 * x_3)$

36 186 $(x_1 * (1-x_2)) + x_3 - (x_1 * (1-x_2)) * x_3$

37 188 $(x_1 * (1-x_2)) + ((1-x_1) * x_2) - (x_1 * (1-x_2)) * ((1-x_1) * x_2) + (x_2 * x_3) - (x_2 * x_3) * (x_1 * (1-x_2)) + ((1-x_1) * x_2) - (x_1 * (1-x_2)) * ((1-x_1) * x_2)$

38 190 $(x_1 * (1-x_2)) + ((1-x_1) * x_2) - (x_1 * (1-x_2)) * ((1-x_1) * x_2) + x_3 - x_3 * (x_1 * (1-x_2)) + ((1-x_1) * x_2) - (x_1 * (1-x_2)) * ((1-x_1) * x_2)$

39 191 $(1-x_1) + (1-x_2) - (1-x_1) * (1-x_2) + x_3 - x_3 * (1-x_1) + (1-x_2) - (1-x_1) * (1-x_2)$

40 194 $(x_1 * x_2) + ((1-x_1) * (1-x_2) * x_3) - (x_1 * x_2) * ((1-x_1) * (1-x_2) * x_3)$

41 200 $(x_1 * x_2) + (x_2 * x_3) - (x_1 * x_2) * (x_2 * x_3)$

42 202 $(x_1 * x_2) + ((1-x_1) * x_3) - (x_1 * x_2) * ((1-x_1) * x_3)$

43 203 $(x_1 * x_2) + ((1-x_1) * (1-x_2)) - (x_1 * x_2) * ((1-x_1) * (1-x_2)) + (x_2 * x_3) - (x_2 * x_3) * (x_1 * x_2) + ((1-x_1) * (1-x_2)) - (x_1 * x_2) * ((1-x_1) * (1-x_2))$

44 206 $((1-x_1) * x_3) + x_2 - ((1-x_1) * x_3) * x_2$

45 224 $(x_1 * x_2) + (x_1 * x_3) - (x_1 * x_2) * (x_1 * x_3)$

46 226 $(x_1 * x_2) + ((1-x_2) * x_3) - (x_1 * x_2) * ((1-x_2) * x_3)$

47 227 $(x_1 * x_2) + ((1-x_1) * (1-x_2)) - (x_1 * x_2) * ((1-x_1) * (1-x_2)) + ((1-x_2) * x_3) - ((1-x_2) * x_3) * (x_1 * x_2) + ((1-x_1) * (1-x_2)) - (x_1 * x_2) * ((1-x_1) * (1-x_2))$

48 232 $(x_1 * x_2) + (x_1 * x_3) - (x_1 * x_2) * (x_1 * x_3) + (x_2 * x_3) - (x_2 * x_3) * (x_1 * x_2) + (x_1 * x_3) - (x_1 * x_2) * (x_1 * x_3)$

49 234 $(x_1 * x_2) + x_3 - (x_1 * x_2) * x_3$

50 235 $(x_1 * x_2) + ((1-x_1) * (1-x_2)) - (x_1 * x_2) * ((1-x_1) * (1-x_2)) + x_3 - x_3 * (x_1 * x_2) + ((1-x_1) * (1-x_2)) - (x_1 * x_2) * ((1-x_1) * (1-x_2))$

51 236 $(x_1 * x_3) + x_2 - (x_1 * x_3) * x_2$

52 239 $(1-x_1) + x_2 - (1-x_1) * x_2 + x_3 - x_3 * (1-x_1) + x_2 - (1-x_1) * x_2$

53 242 $x_1 + ((1-x_2) * x_3) - x_1 * ((1-x_2) * x_3)$

54 248 $x_1 + (x_2 * x_3) - x_1 * (x_2 * x_3)$

55 251 $x_1 + (1-x_2) - x_1 * (1-x_2) + x_3 - x_3 * x_1 + (1-x_2) - x_1 * (1-x_2)$

56 254 $x_1 + x_2 - x_1 * x_2 + x_3 - x_3 * x_1 + x_2 - x_1 * x_2$

Структура рефлексивных булевых функций от трех переменных

Структура Φ_3 представлена на рис.7 и справедлива формула $\Phi_3 = \Phi_{1,3} \vee \Phi_{2,3} \vee \Phi_{3,3}$.

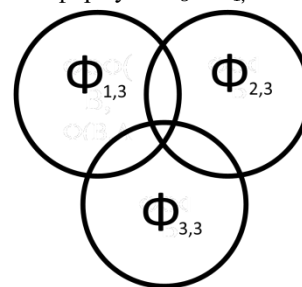


Рисунок 7. – Структура множества Φ_3 .

Пересечение множеств $\Phi_{1,3}$ и $\Phi_{2,3}$ включает следующие функции:

- 1 64 $x_1 \wedge x_2 \wedge !x_3$
- 2 84 $(x_1 \wedge !x_3) \vee (x_2 \wedge !x_3)$
- 3 128 $x_1 \wedge x_2 \wedge x_3$
- 4 168 $(x_1 \wedge x_3) \vee (x_2 \wedge x_3)$
- 5 196 $(x_1 \wedge x_2) \vee (x_2 \wedge !x_3)$
- 6 200 $(x_1 \wedge x_2) \vee (x_2 \wedge x_3)$

7 208 $(x1 \wedge x2) \vee (x1 \wedge !x3)$
 8 212 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (x2 \wedge !x3)$
 9 213 $(x1 \wedge x2) \vee !x3$
 10 216 $(x1 \wedge x2) \vee (x1 \wedge !x3) \vee (x2 \wedge x3)$
 11 220 $(x1 \wedge !x3) \vee x2$
 12 224 $(x1 \wedge x2) \vee (x1 \wedge x3)$
 13 228 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge !x3)$
 14 232 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge x3)$
 15 234 $(x1 \wedge x2) \vee x3$
 16 236 $(x1 \wedge x3) \vee x2$
 17 244 $x1 \vee (x2 \wedge !x3)$
 18 248 $x1 \vee (x2 \wedge x3)$
 19 253 $x1 \vee x2 \vee !x3$
 20 254 $x1 \vee x2 \vee x3$.

Пересечение множеств $\Phi_{1,3}$ и $\Phi_{3,3}$ представлено ниже.

1 32 $x1 \wedge !x2 \wedge x3$
 2 50 $(x1 \wedge !x2) \vee (!x2 \wedge x3)$
 3 128 $x1 \wedge x2 \wedge x3$
 4 162 $(x1 \wedge x3) \vee (!x2 \wedge x3)$
 5 168 $(x1 \wedge x3) \vee (x2 \wedge x3)$
 6 176 $(x1 \wedge !x2) \vee (x1 \wedge x3)$
 7 178 $(x1 \wedge !x2) \vee (x1 \wedge x3) \vee (!x2 \wedge x3)$
 8 179 $(x1 \wedge x3) \vee !x2$
 9 184 $(x1 \wedge !x2) \vee (x2 \wedge x3)$
 10 186 $(x1 \wedge !x2) \vee x3$
 11 200 $(x1 \wedge x2) \vee (x2 \wedge x3)$
 12 224 $(x1 \wedge x2) \vee (x1 \wedge x3)$
 13 226 $(x1 \wedge x2) \vee (!x2 \wedge x3)$
 14 232 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge x3)$
 15 234 $(x1 \wedge x2) \vee x3$
 16 236 $(x1 \wedge x3) \vee x2$
 17 242 $x1 \vee (!x2 \wedge x3)$
 18 248 $x1 \vee (x2 \wedge x3)$
 19 251 $x1 \vee !x2 \vee x3$
 20 254 $x1 \vee x2 \vee x3$

Пересечением множеств $\Phi_{2,3}$ и $\Phi_{3,3}$ является следующий список функций.

1 8 $!x1 \wedge x2 \wedge x3$
 2 14 $(!x1 \wedge x2) \vee (!x1 \wedge x3)$
 3 128 $x1 \wedge x2 \wedge x3$
 4 138 $(!x1 \wedge x3) \vee (x2 \wedge x3)$
 5 140 $(!x1 \wedge x2) \vee (x2 \wedge x3)$
 6 142 $(!x1 \wedge x2) \vee (!x1 \wedge x3) \vee (x2 \wedge x3)$
 7 143 $!x1 \vee (x2 \wedge x3)$
 8 168 $(x1 \wedge x3) \vee (x2 \wedge x3)$
 9 172 $(x1 \wedge x3) \vee (!x1 \wedge x2)$
 10 174 $(!x1 \wedge x2) \vee x3$
 11 200 $(x1 \wedge x2) \vee (x2 \wedge x3)$
 12 202 $(x1 \wedge x2) \vee (!x1 \wedge x3)$
 13 206 $(!x1 \wedge x3) \vee x2$
 14 224 $(x1 \wedge x2) \vee (x1 \wedge x3)$
 15 232 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge x3)$
 16 234 $(x1 \wedge x2) \vee x3$
 17 236 $(x1 \wedge x3) \vee x2$
 18 239 $!x1 \vee x2 \vee x3$

19 248 $x1 \vee (x2 \wedge x3)$
 20 254 $x1 \vee x2 \vee x3$

Элементами пересечением множеств $\Phi_{1,3}$ и $\Phi_{2,3}$, а также $\Phi_{2,3}$ являются функции в следующем списке.

1 128 $x1 \wedge x2 \wedge x3$
 2 168 $(x1 \wedge x3) \vee (x2 \wedge x3)$
 3 200 $(x1 \wedge x2) \vee (x2 \wedge x3)$
 4 224 $(x1 \wedge x2) \vee (x1 \wedge x3)$
 5 232 $(x1 \wedge x2) \vee (x1 \wedge x3) \vee (x2 \wedge x3)$
 6 234 $(x1 \wedge x2) \vee x3$
 7 236 $(x1 \wedge x3) \vee x2$
 8 248 $x1 \vee (x2 \wedge x3)$
 9 254 $x1 \vee x2 \vee x3$

Рефлексивные булевы функции от четырех переменных

На основании компьютерного расчета длительностью 3 часа (использовалось одно ядро I3 4170 с частотой 3.7 ГГц) было определено, что из общего числа всех булевых функций от четырех переменных, равного $2^{16}=65536$, рефлексивными являются 6124.

Ясно, что их все невозможно представить в данной статье. Расчеты для функций от пяти переменных малоосуществимы.

Приведем только несколько из них.

1 2 $!x1 \wedge !x2 \wedge !x3 \wedge x4$
 2 8 $!x1 \wedge !x2 \wedge x3 \wedge x4$
 3 11 $(!x1 \wedge !x2 \wedge !x3) \vee (!x1 \wedge !x2 \wedge x4)$

 6124 65534 $x1 \vee x2 \vee x3 \vee x4$

Применение рефлексивных булевых рассуждений

Рассмотрим общий ход рефлексивных рассуждений Фреге с их достаточно значимым выводом в [6].

Г.Фреге показывает на основе сравнения бинарных признаков свойств сущностей анализа, что существует третья область помимо области представлений человека и области объектов внешнего мира.

Согласно логическим исследованиям Фреге, мысль не является объектом внешнего мира, а также не является внутренним представлением человека, т. е. существует мир мыслей, отдельно от человеческого мира. Так представления не могут быть восприняты сенсорными системами (слух, зрение...) представлениями обладают, т.е. представления требуют существования носителя и имеют только одного носителя. Они разделяют внутренний и внешний мир для человека.

Мысль не является ни внутренним представлением, ни объектом внешнего мира.

Существует 3-я область, элементы которой мысли – совпадают с представлениями, в том отношении, что они не могут быть восприняты сенсорными подсистемами, чувствами.

С объектами внешнего мира мысли совпадают в том, что они не предполагают носителя (не зависят от субъекта).

Пример – мысль, выраженная в теореме Пифагора, является истинной безотносительно ко времени.

Аналогичный подход мы видим при анализе известных языковых высказываний в [9,15]. В [16] рассматривается влияние рефлексивных процессов в эволюции социальных отношений и их эффективности.

Заключение

В работе даны определения рефлексивных булевых функций. Они рассматриваются как базовые примитивы самосознания.

Литература

1.П.Т. де Шарден. Феномен человека.- М.:Наука,1987.-240 с.

2. Выготский Л. С. Сознание как проблема психологии поведения. Собр. сочинений. Т. 1., М.: Педагогика. с. 78 – 98.

3. Выготский Л. С. Мышление и речь. Собр. сочинений. Т. 2. М.: Педагогика, 1982. с. 179.

4. Манин Ю. И. К проблеме ранних стадий речи и сознания (филогенез) // Интеллектуальные процессы и их моделирование. М.: Наука, 1987. с. 154 –178.

5.Лекторский В.А. Субъект, объект, сознание.-М:Наука, 1980, 356 с.

6.Г.Фреге.Мысль: логическое исследование //Философия, логика, язык.- М.:Прогресс,1987.-с.18-47.

7.Андрюхин А.И., Кузнецов А.В. Компьютерное исследование физических аспектов рефлексивности мышления человека//Научные труды ДГТУ.Серия:Проблемы моделирования и автоматизации проектирования динамических систем.Вып. 29.2002 г.,с.218-226.

8.Хинтика Я. Логико-эпистемологические исследования. М.: Прогресс, 1980. с. 446

9. Андрюхин А. И., Кузнецов А. В. Булевы модели самодиагностирования дискретных систем // Известия №1 ДонТу-Таганрог, апрель 2001 года. Материалы 2-го международного научно-технического семинара

В работе определены классы рефлексивных булевых функций и выполнена их классификация. Приведены все рефлексивные булевы функции от двух и трех переменных. Также вычислены все рефлексивные булевы функции от четырех переменных.

Представлены действительные функции являющиеся расширениями рефлексивных булевых функций множества $\Phi_{3,3}$.

Показано, что логическая операция импликации принадлежит к классу рефлексивных функций.

Дальнейшая перспектива исследований связана с так называемыми «парадоксальными» рефлексивными булевыми функциями[12-14].

«Практика и перспективы развития институционального партнёрства» с. 168-175.

10. Г.Н.Борисюк и др.Осцилляторные нейронные сети. Математические результаты и приложения// Математическое моделирование, 1992. - № 1. - С.3-43.

11.Vladimir A. Lefebvre. The Law of Self-Reflexion: A Possible Unified Explanation for the Three Different Psychological Phenomena. Uni\ersity of California, IrVine USA

12.Андрюхин А.И. Оценка рефлексивных связей в вероятностной логике//Системный анализ в науках о природе и обществе, Донецк, Доннту, №1(4)-2(5)'2013, с.75-80.

13.Андрюхин А. И. Вероятностные оценки рефлексивных логических связей / А. И. Андрюхин // Наукові праці Донецького національного технічного університету. Серія : Інформатика, кібернетика та обчислювальна техніка. - 2015. - Вип. 1. - С. 147-154. - Режим доступу: http://nbuV.gov.ua/j-pdf/Npdntu_inf_2015_1_25.pdf.

14. Андрюхин А.И., Подтынный С.Д. КМОП-реализации рефлексивных систем вероятностной логики//Информатика и кибернетика,№ 1, 2015,с.25-34.

15. Андрюхин А.И. Модельные представления антиномии в булевых сетях // Искусственный интеллект, 1998,N 1, с. 35–4

16.Лепский В.Е., Зорина Г.И. Рефлексивное предприятие XXI ве-

ка//Рефлексивные процессы и управление, № 2, 2005, том 5, с.21-401.

Andryukhin AI Reflexive Boolean functions. This work refers to the fundamental problem of determining the basic primitive consciousness. The human brain based on its characteristics, is able to self-organize and adapt in the environment. Classes reflexve Boolean functions defined in the work. Their classification is made. Also, all reflective Boolean functions of two and three variables are given. All reflective Boolean functions of four variables are defined. Results of computer experiments are given. It is shown that the logical implication operation belongs to the class of reflective functions. Software system Wolfram Mathematics used in the calculations.

Keywords: reflection, Boolean functions, self-organization, consciousness, brain

Андрюхин А.И. Рефлексивные булевы функции. Эта работа относится к фундаментальной проблеме определения базовых примитивов самосознания. Человеческий мозг основываясь на их свойствах, способен самоорганизовываться и адаптироваться во внешней среде. В работе определены классы рефлексивных булевых функций. Выполнена их классификация. Все рефлексивные булевы функции от двух и трех переменных приведены. Также все рефлексивные булевы функции от четырех переменных определены. Показано, что логическая операция импликации принадлежит к классу рефлексивных функций. Результаты компьютерных экспериментов приведены. Программная система Вольфрам Математика использовалась в расчетах.

Ключевые слова: рефлексия, булевы функции, самоорганизация

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук С.Н. Судаковым

УДК 004.7

Анализ UDP-трафика

Бельков Д.В., Едемская Е.Н.
Донецкий национальный технический университет
belkov65@list.ru

Бельков Д.В., Едемская Е.Н. Анализ UDP-трафика. В данной работе для реализаций процесса задержки и джиттера UDP-пакетов объемом 64 и 512 байт выполнен анализ плотности распределения, автокорреляционных функций и энергетических спектров. Найдены значения показателя Херста H . Получены следующие результаты. Изучаемые временные ряды не подчиняются нормальному распределению, ряды не обладают экспоненциально спадающей АКФ, свойственной случайным рядам. Для процесса задержки пакетов объемом 64 байт выполняется условие $0,5 < H < 1$, для пакетов объемом 512 байт выполняется условие $0 < H < 0,5$. Для джиттера при передаче пакетов объемом 64 байт и 512 байт показатель Херста близок к значению 0,5 – ряды являются случайными.

Ключевые слова: UDP-пакеты, трафик, плотность распределения, автокорреляционная функция, энергетический спектр, показатель Херста.

Введение

Многочисленные исследования [1-5] трафика компьютерных сетей свидетельствуют о том, что он обладает свойством масштабной инвариантности - имеет особую фрактальную (самоподобную) структуру, сохраняющуюся на разных масштабах. В процессе передачи возникают большие всплески при относительно низком среднем уровне трафика. Данное явление, которое значительно влияет на потери, задержки и джиттер пакетов, не учитывается в классической теории телетрафика. Расчет параметров системы распределения информации, предназначенной для обработки сетевого трафика, по классическим формулам дает некорректные, неоправданно оптимистические результаты. Алгоритмы обработки трафика, созданные для работы с простейшим потоком неэффективны для фрактальных потоков.

Ситуация, сложившаяся в современных компьютерных сетях, наличие большого количества сетевых маршрутов на которых периодически возникают резкие колебания задержки в передаче данных и большие потери пакетов, появление новых свойств сетевого трафика, необходимость обеспечения высокого качества обслуживания приложений, делают актуальным статистический анализ сетевого трафика.

Целью настоящей работы является исследование структуры трафика, направленное на выявление его характерных особенностей. Изучается один из основных сетевых процессов – процесс задержки пакетов, который служит для получения информации о состоянии сети методом “черного ящика”, когда через сеть

пропускается последовательность пакетов, и на основании времени их прохождения до удаленного узла делаются выводы о загрузке сети.

В работе решаются следующие задачи: оцениваются плотности распределения, автокорреляционные функции (АКФ), энергетические спектры изучаемых временных рядов, определяется степень фрактальности (показатель Херста). Исследования выполнены в среде MatLab.

Статистический анализ трафика

Для изучения выбраны две реализации сетевого трафика [6], полученные в 2007 году в университете города Наполи (Италия). Согласно лицензии данные свободно доступны для анализа. Измерения проводились каждые 10 миллисекунд, получено свыше 12 000 отсчетов. В первом случае (ряды UDP_d64 и UDP_j64) измерялись задержка и джиттер UDP-пакетов объемом 64 байт, во втором случае (ряды UDP_d512 и UDP_j512) измерялись задержка и джиттер UDP-пакетов объемом 512 байт. Отправитель имел ADSL-доступ (640 Kbps), на стороне получателя - 100 Mbps Ethernet, операционная система Linux на каждой из сторон, скорость передачи 100 pps.

Необходимо отметить, что для процесса задержки на маршруте прохождения пакета можно выделить постоянную минимальную составляющую, обусловленную отсутствием очередей и переменную составляющую, возникающую из-за задержек в очередях. Задержка измеряется на стороне источника. Она складывается из времени обработки, времени распространения, времени коммутации, времени

ожидания в очереди. Изучаемые временные ряды показаны на рисунках 1-4.

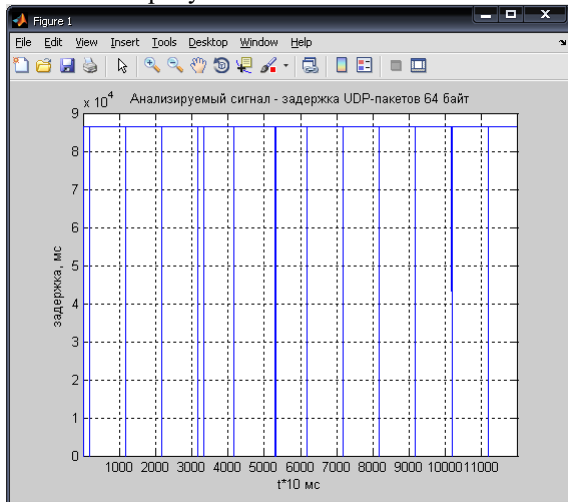


Рисунок 1 - Задержка при передаче пакетов объемом 64 байт

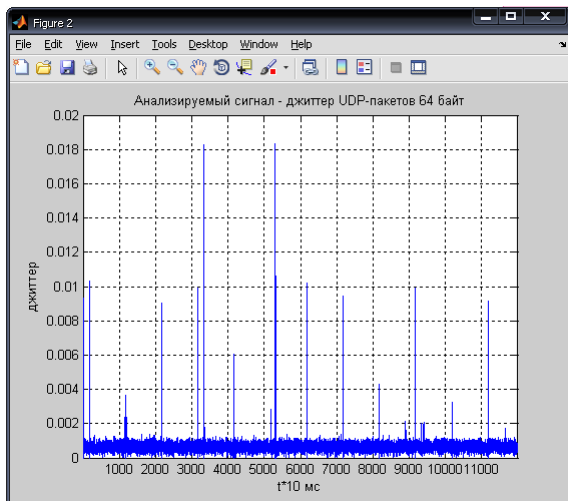


Рисунок 2 - Джиттер при передаче пакетов объемом 64 байт

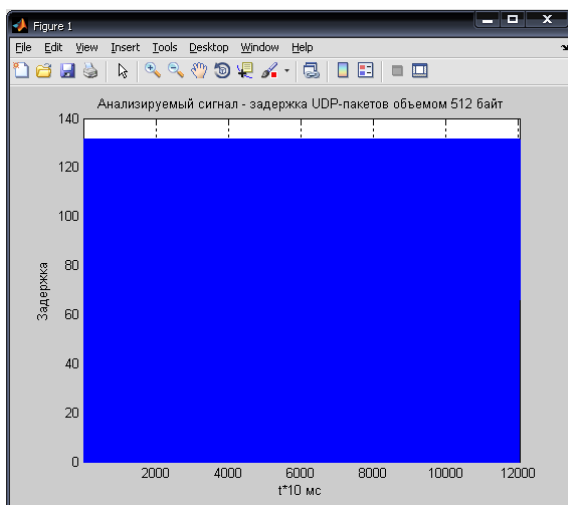


Рисунок 3 - Задержка при передаче пакетов объемом 512 байт

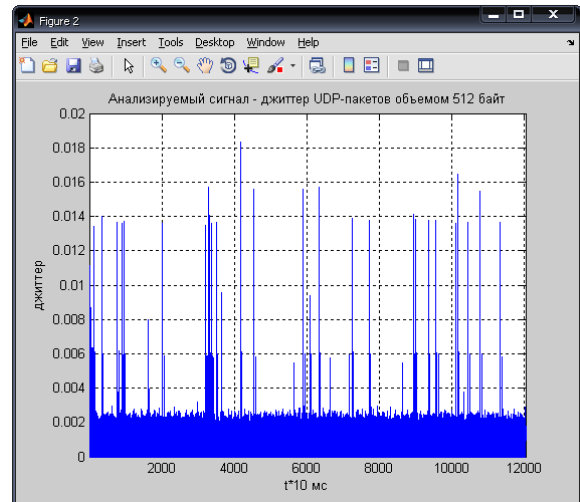


Рисунок 4 - Джиттер при передаче пакетов объемом 512 байт

Оценка плотностей распределения проводится на основании гистограмм относительных частот, показанных на рисунках 5-8.

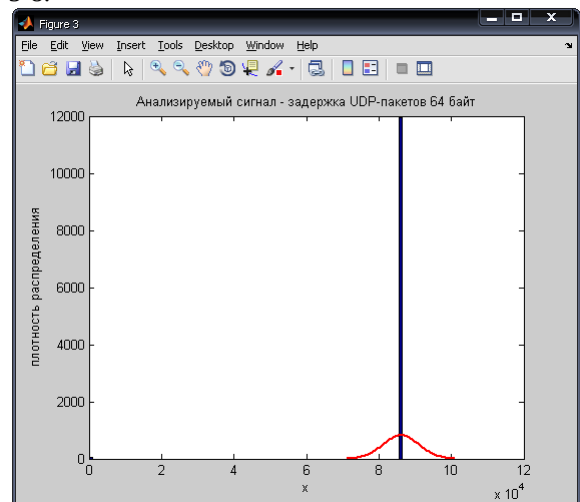


Рисунок 5 - Плотность распределения временного ряда UDP_d64

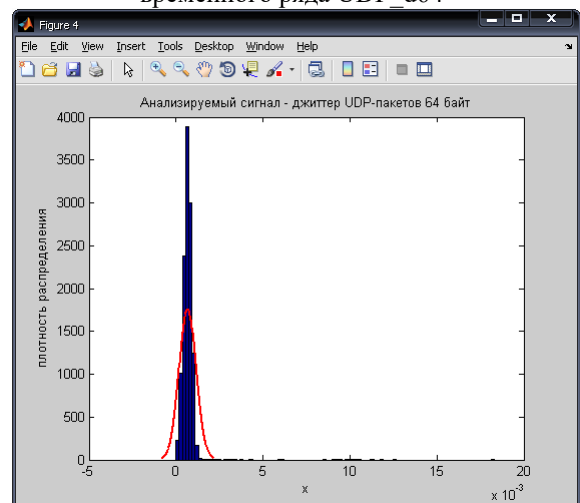


Рисунок 6 - Плотность распределения временного ряда UDP_j64

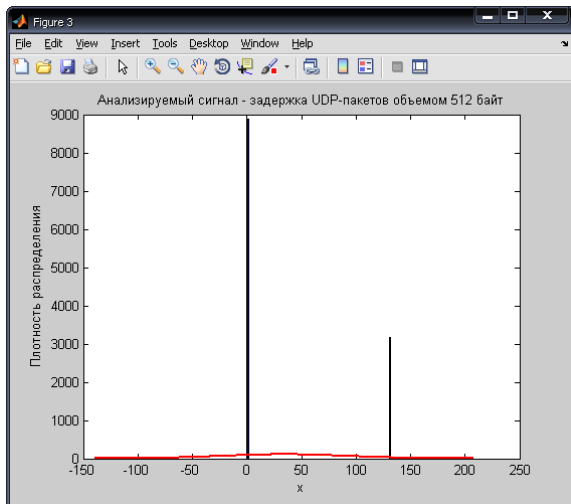


Рисунок 7 - Плотность распределения временного ряда UDP_d512

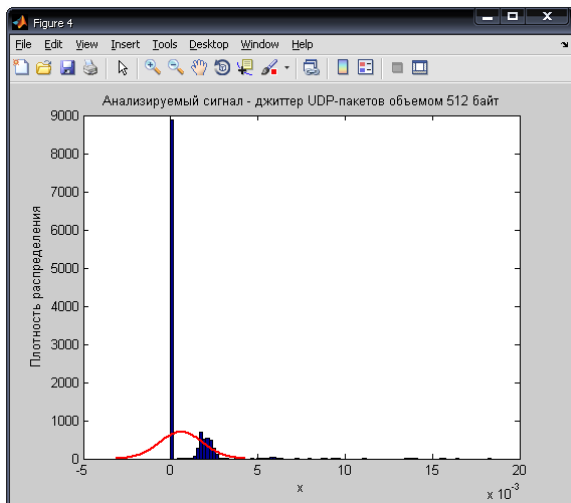


Рисунок 8 - Плотность распределения временного ряда UDP_j512

Временные ряды UDP_d64, UDP_d512, UDP_j512 не подчиняются нормальному распределению. Ряд UDP_j64 близок к случайному.

В работе необходимо определить обладают ли временные ряды медленно убывающей зависимостью или быстро убывающей зависимостью.

Процесс X обладает медленно убывающей зависимостью (МУЗ) [long-range dependence], если для его АКФ выполняется условие $r(k) \sim k^{-\beta} \cdot L_1(k) + C_1, k \rightarrow \infty$, где $C_1 = \text{const}$, $0 < \beta < 1$ и L_1 - медленно меняющаяся на бесконечности функция:

$$\lim_{t \rightarrow \infty} \frac{L_1(tx)}{L_1(t)} = 1 \text{ для всех } x. \text{ Процессы с МУЗ}$$

характеризуются автокорреляционной функцией, которая убывает по степенному закону при увеличении временной задержки (лага) [7]. В

отличие от процессов с МУЗ, процессы с быстро убывающей зависимостью (БУЗ) [short-range dependence] обладают экспоненциально спадающей АКФ вида $r(k) \sim \rho^k, k \rightarrow \infty$.

Вычисления АКФ по формуле

$$r(k) = \frac{\sum_{i=1}^{N-\tau} (X_i - \bar{X})(X_{i+k} - \bar{X})}{(N - \tau)\sigma^2(X)}, \quad \text{где } \bar{X} -$$

выборочное среднее ряда X , $\sigma^2(X)$ - выборочная дисперсия ряда X , $k=0,1,\dots$. Автокорреляционные функции исследуемых временных рядов приведены на рисунках 9-12. Они не обращаются в ноль при больших значениях k , что говорит о медленном убывании АКФ и присутствии МУЗ во всех исследуемых реализациях трафика.

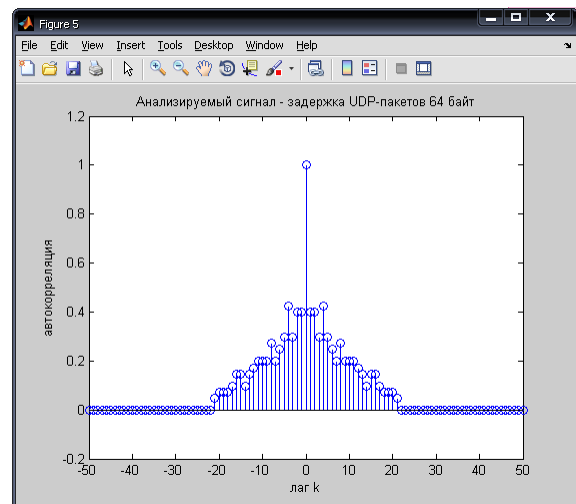


Рисунок 9 - АКФ временного ряда UDP_d64

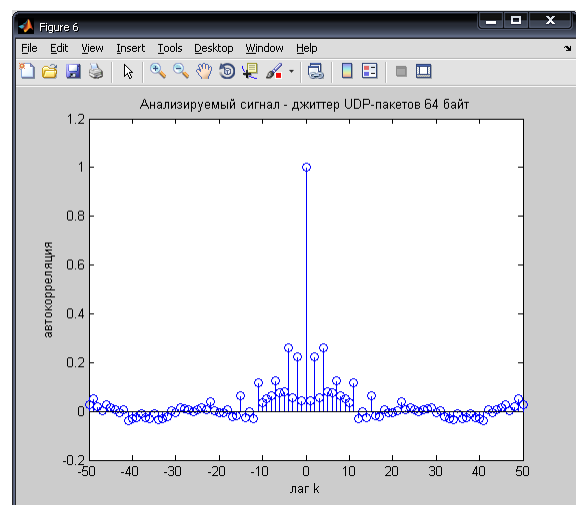


Рисунок 10 - АКФ временного ряда UDP_j64

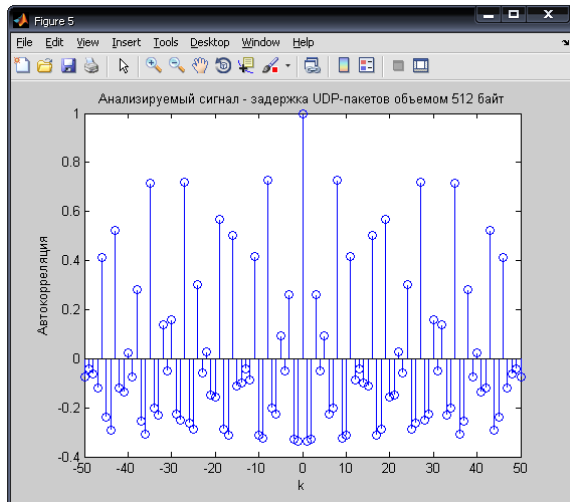


Рисунок 11 - АКФ временного ряда UDP_d512

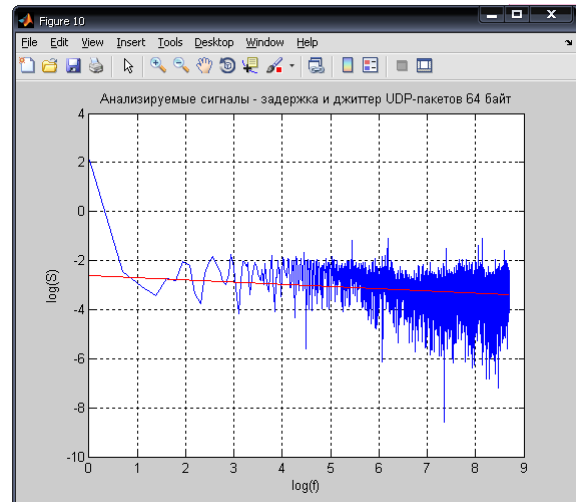
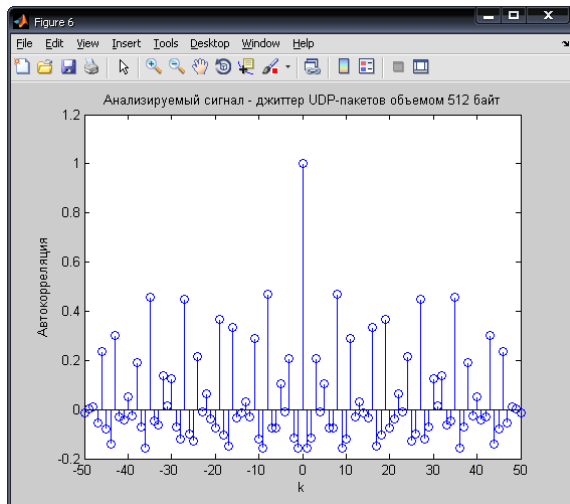
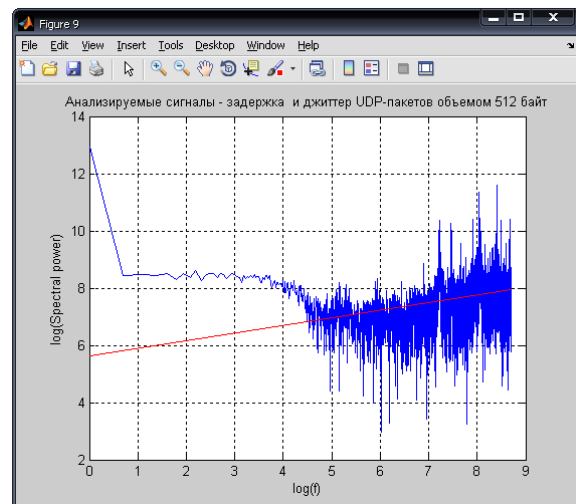
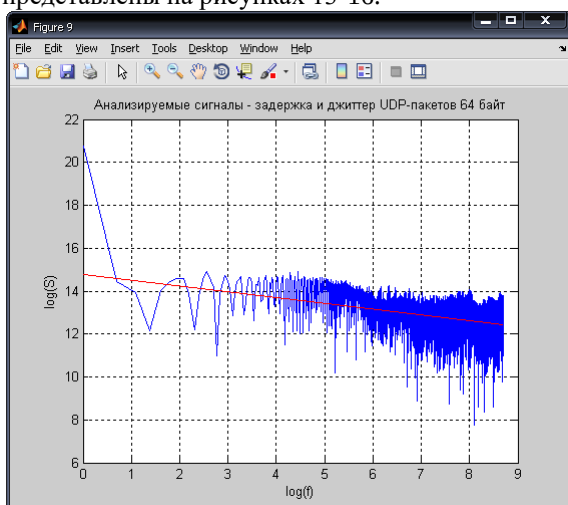
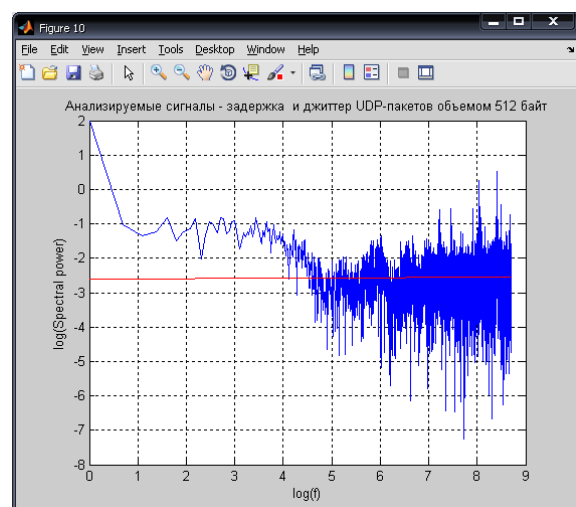
Рисунок 14 - Спектральная плотность
временного ряда UDP_j64

Рисунок 12 - АКФ временного ряда UDP_j512

Рисунок 15 - Спектральная плотность
временного ряда UDP_d512

Известно [8-10], что в частотной области МУЗ приводит к степенному закону поведения спектральной плотности процесса. Энергетические спектры временных рядов представлены на рисунках 13-16.

Рисунок 13 - Спектральная плотность
временного ряда UDP_d64Рисунок 16 - Спектральная плотность
временного ряда UDP_j512

Функция Херста

В этом разделе оценивается показатель Херста H , который является мерой длительности долгосрочной зависимости процесса [4]. Значение $H=0,5$ указывает на отсутствие долгосрочной зависимости. Чем ближе значение H к 1, тем выше степень устойчивости долгосрочной зависимости. При $0 \leq H < 0,5$ временной ряд является трендонеустойчивым (антиперсистентным). Он более изменчив, чем случайный ряд, поскольку состоит из частых реверсов спад-подъем. При $0,5 < H \leq 1$ ряд трендоустойчив. Тенденция его изменения может быть спрогнозирована.

В работе [11] для анализа трафика предложено использовать функцию Херста. Ее график строится следующим образом. По оси абсцисс откладываем значения $N=2,3,\dots,N_m$, где N_m – максимальная длина временного ряда, N – текущее количество элементов временного ряда. Для каждого значения N с помощью R/S-анализа определяем величину показателя Херста $H(N)$ и откладываем ее по оси ординат. Показатель Херста каждого ряда указан в таблице 1.

1. Таблица 1. Показатель Херста

Временной ряд	Показатель Херста
UDPd64	0,66
UDPj64	0,49
UDPd512	0,37
UDPj512	0,52

Графики функции Херста изучаемых рядов показаны на рисунках 17-20.

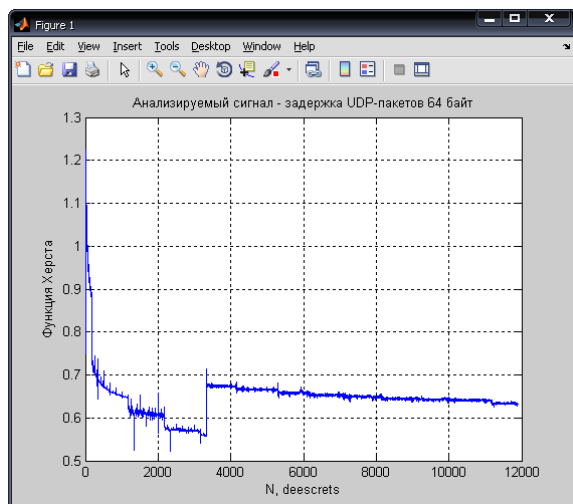


Рисунок 17 – Функция Херста ряда UDP_d64

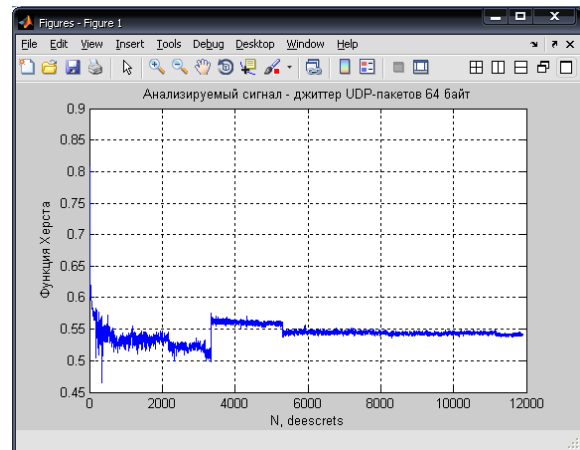


Рисунок 18 – Функция Херста ряда UDP_j64

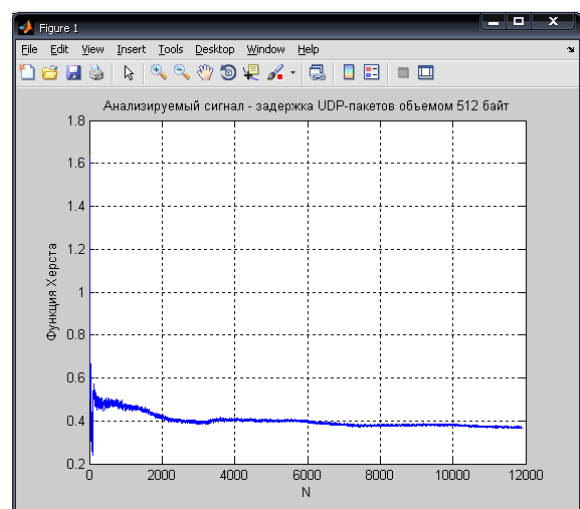


Рисунок 19 – Функция Херста ряда UDP_d512

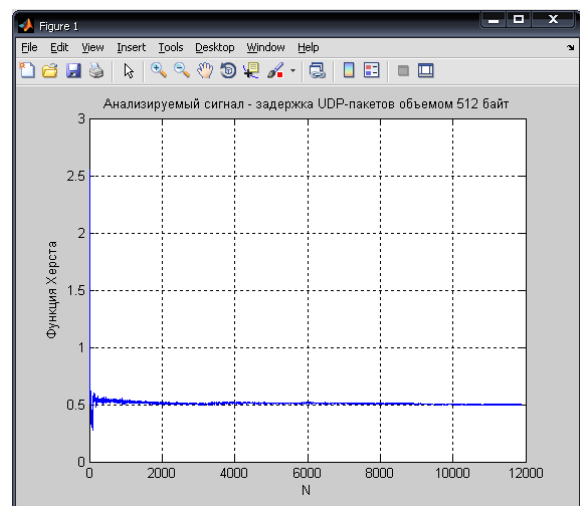


Рисунок 20 – Функция Херста ряда UDP_j512

Выводы

Современный трафик имеет особенности, затрудняющие применение теории очередей. Для процессов передачи данных пакетным трафиком характерно обнаруженное на практике свойство фрактальности.

В данной работе для реализаций [3] процесса задержки и джиттера UDP-пакетов объемом 64 байт выполнен анализ плотности распределения, автокорреляционных функций и энергетических спектров. Найдены значения показателя Херста H . Получены следующие результаты:

2. Изучаемые временные ряды не подчиняются нормальному распределению.
3. Изучаемые временные ряды не обладают экспоненциально спадающей АКФ, свойственной случайным рядам.
4. Для процесса задержки пакетов объемом 64 байт выполняется условие $0,5 < H < 1$. Этот ряд является трендоустойчивым (персистентным) и обладает долговременной памятью.
5. Для процесса задержки пакетов объемом 512 байт выполняется условие $0 < H < 0,5$. Этот ряд является антиперсистентным.
6. Для джиттера при передаче пакетов объемом 64 и 512 байт показатель Херста близок к значению 0,5 – ряды являются случайными.

Перспективным направлением является создание математических моделей сетевых процессов.

Литература

1. Park K. Self-Similar Network Traffic: An Overview. [Электронный ресурс], 2003. – Режим доступа: <http://pi.314159.ru/park1.pdf>
2. Willinger W., Taqqu M.S., Errimilli A. A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks. [Электронный ресурс], 2001. – Режим доступа: <http://linkage.rockefeller.edu/wli/reading/taqqu96.pdf>
3. Иванов А. В. Разработка и исследование алгоритмов прогнозирования и управления очередями в компьютерных сетях. Санкт – Петербург. – 2001. – 18 с.
4. Осин А.В. Влияние самоподобности речевого трафика на качество обслуживания в телекоммуникационных сетях. Автореферат диссертации. Москва. – 2005. – 20 с.
5. Ложковський А.Г. Аналіз і синтез систем розподілу інформації в умовах мультисервісного трафіка. Автореферат дисертації. – Одеса, 2010. – 38 с.
6. Network tools and traffic traces. [Электронный ресурс], 2007. – Режим доступа: <http://www.grid.unina.it/Traffic/Traces/ttraces.php>
7. Петров В.В. Структура телетрафика и алгоритм обеспечения качества обслуживания при влиянии эффекта самоподобия. Автореферат диссертации. Москва. – 2004. – 20 с.
8. Trang Dinh Dang New results in multifractal traffic analysis and modeling. Budapest, – 2002. – 101 с.
9. Шепухин О.И., Тенякшев А.М., Осин А.В. М. Фрактальные процессы в телекоммуникациях Москва: Радиотехника, - 2003.- 480с.
10. Городецкий А. Я., Заболоцкий В. С. Фрактальные процессы в компьютерных сетях. – СПб: Издательство СПбГТУ, 2000. – 139 с.
11. Бельков Д.В., Едемская Е.Н. Исследование сетевого трафика с помощью функции Херста. Информатика и кибернетика № 2. Донецк: ДонНТУ, 2015. - С.39-46.

Бельков Д.В., Едемская Е.Н. Анализ UDP-трафика. В данной работе для реализаций процесса задержки и джиттера UDP-пакетов объемом 64 и 512 байт выполнен анализ плотности распределения, автокорреляционных функций и энергетических спектров. Найдены значения показателя Херста H . Получены следующие результаты. Изучаемые временные ряды не подчиняются нормальному распределению, ряды не обладают экспоненциально спадающей АКФ, свойственной случайным рядам. Для процесса задержки пакетов объемом 64 байт выполняется условие $0,5 < H < 1$, для пакетов объемом 512 байт выполняется условие $0 < H < 0,5$. Для джиттера при передаче пакетов объемом 64 байт и 512 байт показатель Херста близок к значению 0,5 – ряды являются случайными.

Ключевые слова: UDP-пакеты, трафик, плотность распределения, автокорреляционная функция, энергетический спектр, показатель Херста.

Belkov D.V., Edemskay E.N. Analysis of UDP-traffic. In this work for realization delay process and jitter process of UDP-packets by volume of a 64 byte and 512 byte the analysis of closeness of distributing, autocorrelation functions and power spectrums are executed. The values of Hurst H index are found. Next results are got. The studied time series do not submit to normal distribution, series do not possess exponentially falling ACF, incident to the casual series. For the delay process of packets by volume of a 64 byte the condition $0,5 < H < 1$ is executed, for the delay process of packets by volume of a 512 byte the condition $0 < H < 0,5$. For jitter process at the transmission of packets by volume of a 64 byte and 512 byte the Hurst indexes are near to the value 0,5 - the series are stochastic.

Keywords: UDP-packets, traffic, closeness of distributing, autocorrelation function, power spectrum, the Hurst index.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 004.652.4

Разработка информационной системы истории болезни органов брюшной полости пациентов

А.В. Борота, А.А. Борота, А.П. Кухто, Н.К. Базиян-Кухто
Донецкий национальный медицинский университет
naira-251088@mail.ru

О.И. Федяев, В.С. Бакаленко
Донецкий национальный технический университет
fedyaev@donntu.org, valeriy.bakalenko@gmail.com

Борота А.В., Борота А.А., Кухто А.П., Базиян-Кухто Н.К., Федяев О.И., Бакаленко В.С. *Разработка информационной системы истории болезни органов брюшной полости пациентов.* В статье рассматривается разработка медицинской информационно-справочной системы, автоматизирующей процессы сбора, хранения, редактирования и предварительной статистической обработки данных для предоставления медицинским работникам необходимой информации о пред- и постоперационных периодах лечения болезней органов брюшной полости пациентов. Для решения данной задачи использовались современные методы и технологии построения информационных систем с базой данных реляционного типа. В результате построена объектно-ориентированная система открытого типа, которая в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений с целью эффективного лечения заболеваний желудочно-кишечного тракта.

Ключевые слова: база данных, информационная система, UML-модели, медицина, органы брюшной полости.

Введение

Важной научной программой, направленной на информационное обеспечение здравоохранения, является разработка информационных ресурсов и сервисов, обеспечивающих поддержку принятия решений в сфере здравоохранения [1]. При создании таких систем в первую очередь должны быть реализованы следующие основные процессы: ввод, хранение, обработка и вывод медицинской информации в удобном для пользователя виде. Автоматизация этих процессов, учитывая первичный характер обработки данных, успешно реализуются информационно-справочными системами, которые относятся к первому (нижнему) уровню в архитектурной иерархии информационных систем. В данной статье рассматривается разработка медицинской информационно-справочной системы как базы, которая в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений в направлении лечения заболеваний желудочно-кишечного тракта, как традиционными методами, так и симулятивными операционными вмешательствами (СОВ) [2,3].

За период с 2012 по 2016 год в клинике

общей хирургии №1 Донецкого национального

медицинского университета им. М. Горького на базе Донецкого колопроктологического центра пролечено 189 пациентов с диагнозом язвенного колита (ЯК), из которых 68 (35,9%) были оперированы в один или несколько этапов с последующим реконструктивно-восстановительным вмешательством. Проблема одновременной хирургической коррекции нескольких заболеваний интересует хирургов в течение нескольких десятилетий. Еще в 1985 году Всемирной Организацией Здравоохранения опубликованы статистические данные, согласно которым у 25%-30% больных, подлежащих оперативному лечению в связи с каким-либо заболеванием, выявляются дополнительно одно или несколько заболеваний, требующих одномоментного хирургического лечения. Однако, несмотря на наличие реальных возможностей для оказания полноценной медицинской помощи больным данной категории и достижения максимального медико-социального эффекта, до настоящего времени симультанные оперативные вмешательства производятся лишь у 1,5%-6% этих пациентов [2,3]. Крайне незначительное количество выполняемых сочетанных операций, не

соответствующее реальным потребностям в них, объясняется различными авторами целым рядом причин: неполное обследование больных в предоперационном периоде, недостаточная интраоперационная ревизия органов брюшной полости (ОБП), преувеличение степени операционного риска при определении возможности [3,4].

Предметом дискуссии в настоящее время продолжают оставаться вопросы, связанные с классификацией СОВ, определением показаний и противопоказаний к их выполнению, выбором оперативного доступа, объемом выполнения при поражении смежных органов, последовательностью выполнения основного и симультанных этапов, оценкой и особенностями постагрессивных реакций в послеоперационном периоде, ведением послеоперационного периода, а также социально-экономической эффективностью сочетанных оперативных вмешательств.

Разработка оптимальных алгоритмов диагностики и эффективных методов лечения пациентов с заболеванием толстой кишки (ЗТК) и сочетанной патологией органов брюшной полости (ОБП) является актуальной проблемой колопроктологии. Все это определяет перспективность исследований, направленных на усовершенствование диагностики и лечения данной категории пациентов. Отсутствие единой информационной базы данных о результатах проведенных операций и автоматизированных средств интеллектуального анализа данных затрудняют объективную оценку СОВ и препятствуют их более широкому применению в клинической практике [5-7].

Поэтому цель исследования состоит в разработке и оценке эффективности информационной системы, автоматизирующей процессы сбора, хранения, отсеивания и предварительной статистической обработки данных для предоставления хирургам необходимой информации о пред- и послеоперационных периодах выполнения СОВ у пациентов с различными заболеваниями толстой кишки и хирургической патологией органов брюшной полости.

Предметная область исследования

Задача исследования состоит в анализе результатов лечения объекта исследования, в роли которого выступают сочетанные заболевания толстой кишки (ЗТК) и органов брюшной полости (ОБП). Осложнения со стороны желудочно-кишечного тракта (ЖКТ) создают угрозу для жизни пациента. Поэтому эти осложнения являются абсолютным показанием к хирургическому лечению болезни ЖКТ. Но проблема состоит в том, что на данном этапе

развития медицины нет методики, которая достоверно снижала бы риск рецидива такого типа заболевания. Рассмотрим в качестве примера болезнь Крона (БК) [9]. Хирургическое лечение БК - проблема, которая волновала хирургов всего мира долгое время. Причин этому несколько: увеличение количества случаев заболеваний, отсутствием единого мнения в вопросах хирургической тактики, качество жизни больных после операции. В хирургическом лечении БК сейчас применяются реконструктивно - восстановительные операции. В этом случае очень важно доказать хирургу, используя имеющиеся данные о результатах выполнения пластических операций при БК, целесообразность выполнения реконструктивно - восстановительных операций у данной категории больных. Эта задача может эффективно решаться на основе анализа больших объемов медицинских данных путём использования соответствующих информационных систем.

Предметом исследования являются методы и технологии построения информационных систем с базой данных реляционного типа [10]. Информационная система для анализа лечения толстой кишки (ИС АЛТК) использует реляционную модель данных. Предметная область (медицинские данные о результатах лечения толстой кишки) представлена в виде реляционной модели данных, которая описывается отношением R со схемой γ , являющимся подмножеством кортежей декартового произведения $D_1 \times D_2 \times \dots \times D_n$

$$\{d_1, d_2, \dots, d_n, d_1 \in D_1, d_2 \in D_2, \dots, n \in D_n\}.$$

Домены D_i определяют множество возможных значений атрибута A_i реляционной таблицы. Схема γ отношения R определяется списком имён атрибутов, т.е. названиями колонок таблицы

$$r = \{A_1(D_1), A_2(D_2), \dots, A_n(D_n)\},$$

где $A_1(D_1), A_2(D_2), \dots, A_n(D_n)$ - имена атрибутов (ФИО пациента, пол, дата рождения и т. д.). В ИС АЛТК насчитывается более 70 атрибутов (арность $n > 70$), которые разбиты на четыре группы:

- информация о пациенте;
- информация об операциях пациента;
- информация о гистологических данных пациента;
- информация о результатах различных анализов.

Условия отображения рассматриваемой предметной области в кортежи значений атрибутов $\langle d_1, d_2, \dots, d_n \rangle$ описываются предикатом $P(d_1, d_2, \dots, d_n)$. Конкретные значения кортежей предоставила кафедра общей хирургии №1 ГОУ ВПО «Донецкий национальный медицинский университет им. М.Горького». Таким образом, любой операционный запрос к

базе данных результатов лечения толстой кишки (БД РЛТК) может быть сформирован в виде последовательности, составленной из 8 основных операций реляционной алгебры: объединение, разность, декартово произведение, проекция, селекция и др.

Логические модели информационной системы

На этапе проектирования ИС АЛТК использовался объектно-ориентированный подход. Модели системы описывались на визуальном языке моделирования UML [10, 11]. Концептуальное описание ИС АЛТК представлено диаграммой вариантов использования, которая описывает функциональное назначение системы или, другими словами, то, что система будет делать в процессе своего функционирования (см. рис.1). Медицинские работники, как пользователи системы, представлены одним актёром «Медицинский работник». Система предоставляет актёру две главные функции: создать пациента и просмотреть пациента, вызов которых возможен благодаря двум ассоциативным связям.

Режим ввода информации о пациенте декомпозирован на четыре подфункции, которые подключаются связями <include> к базовому прецеденту «Создать клиента». Каждая подфункция предусматривает ввод одной из групп показателей пациента: общая информация, информация о проведенных операциях, информация об обследованиях и информация о гистологических ответах.

Актёр (пользователь) в режиме просмотра данных о пациенте может редактировать информацию о пациенте или удалить её. Эти возможности на диаграмме прецедентов визуализируются двумя связями расширения <extend>. Режим редактирования информации также предполагает отдельный доступ к каждой группе данных в БД РЛТК по желанию пользователя, что визуализируется связями <extend>.

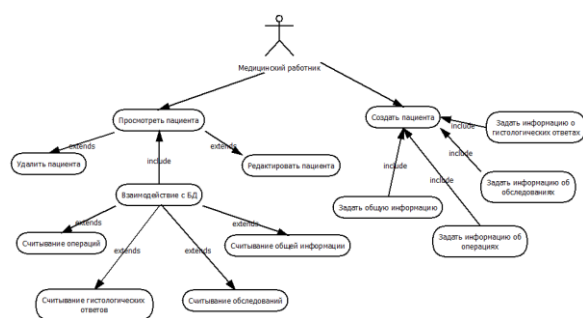


Рисунок 1 – Диаграмма прецедентов

информационно-справочной системы с базой данных результатов лечения органов брюшной полости

Объектно-ориентированный анализ потоков управления каждого прецедента на рис.1 позволил составить логическую структуру информационной системы, которая представлена на рис.2.

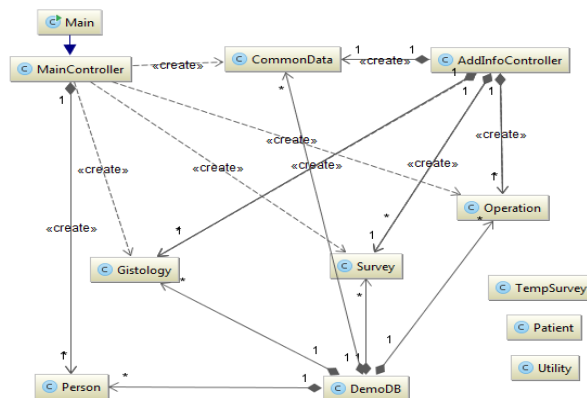


Рисунок 2 – Диаграмма классов информационной системы анализа лечения органов брюшной полости

Типовая декомпозиция выявила 12 классов, между которыми предусмотрены различные отношения (ассоциации, зависимости и агрегации), учитывающие специфику взаимодействия классов. Количественные отношения между классами могут быть двух видов: «один к одному» и «один ко многим».

Класс Main является основным классом, который запускает в работу информационную систему.

Класс MainController отвечает за отображение главного окна приложения со всеми пациентами и полями поиска. Он также считывает из базы данных информацию и обрабатывает её, а также выводит информацию по пациентам в окно с помощью вспомогательного класса Person (поля класса содержат основную информацию о пациенте: ФИО, диагноз и операции).

Класс AddInfoController отвечает за отображение информации в окне пациента и за сохранение информации о пациенте в БД. Сохранение и отображение осуществляется с помощью вспомогательных классов: Operation (содержит информацию об операции), Survey (содержит информацию об обследованиях), Gistology (содержит информацию о гистологическом ответе).

Класс CommonData содержит в своих атрибутах общую информацию о пациенте, а именно: ФИО, адрес, пол, возраст и т.д.

Класс DemoDB – служебный, используется для локальных целей. Он собирает в одном месте всю информацию о пациенте.

Классы Utility, Patient и TempSurvey являются вспомогательными.

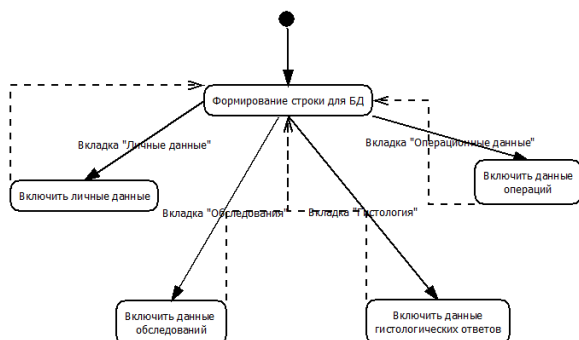


Рисунок 3 – Диаграмма состояний объекта класса AddInfoController

Жизненный цикл объекта AddInfoController описывается диаграмма состояний на рис.3. Названия состояний отражают деятельность объекта класса в каждом из них. В каждом состоянии объект заполняет часть записи соответствующими данными пациента, которая потом сохраняется в файле.

Имеющаяся при системе база данных позволяет в виде реляционных таблиц хранить сведения о пациентах: личные данные о пациента, данные обследования, данные об операциях и данные анализов. Опишем основные таблицы базы данных (см. рис.4). Таблица Patients хранит основную информацию о пациенте (ФИО, адрес, возраст, инвалидность и т.д.). Таблица Operations хранит информацию об операциях пациента (см. рис.5). Она содержит название операции, осложнения, возникшие при операции, тип наркоза, длительность операции, дату и тип операции. Таблица Gists хранит информацию о гистологических ответах пациента. Таблица Survey хранит информацию об обследованиях: типы обследований, дату, параметры исследований (анализы).

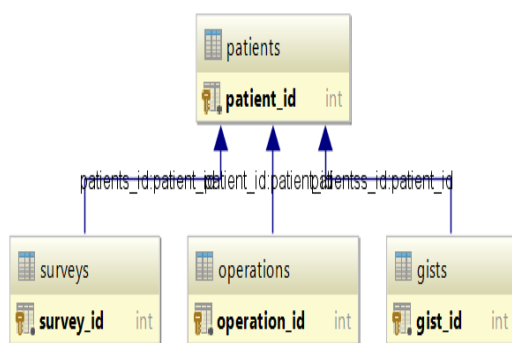


Рисунок 4 – Основные таблицы базы данных информационной системы

operations	
operation_id	int
patient_id	int
operation_text	varchar(255)
operation_complications	varchar(255)
operation_narcoz	varchar(20)
operation_last	smallint
operation_date	date
operation_type	varchar(20)

Рисунок 5 – Структура таблицы о перенесённых пациентом операциях

Диаграмма классов и базы данных сделана средствами IDE IntelliJ IDEA. Остальные диаграммы созданы в бесплатном приложении для создания диаграмм – Dia [12].

Программная реализация системы

Программная реализация информационной системы выполнена на языке программирования Java в инструментальной среде IDE IntelliJ IDEA. Программные компоненты системы располагаются в различных *.java файлах. Интеграция с базой данных осуществлена с помощью драйвера JDBC [12,13].

Построенная по открытому принципу система предусматривает возможность наращивания новых функциональных возможностей. Она проста в применении и не требует профессиональных компьютерных навыков для общения с ней. Для работы информационной системы необходимо установить на компьютере систему программирования Java и любую СУБД MySQL.

Возможности системы заключаются в следующем:

- ввод и коррекция информации о пациентах;
- удаление информации о пациентах;
- получение различной выходной информации по запросам медицинского работника.

Информационная система имеет многооконный пользовательский интерфейс. Интерфейс написан на JavaFX с использованием шаблона MVC. Весь дизайн программной системы описан в файле с расширением *.FXML.

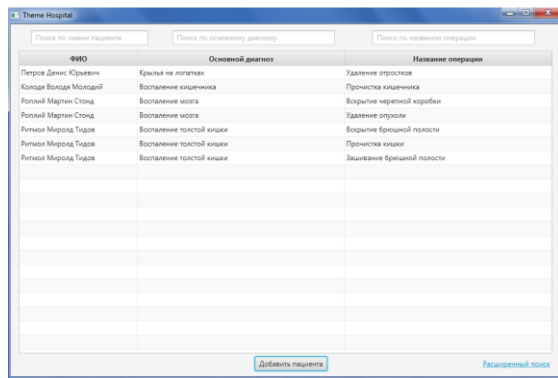


Рисунок 6 – Главное окно пользовательского интерфейса информационной системы

Пользовательский интерфейс проектировался с учётом удобства работы с системой. Достоинствами интерфейса являются: отображение исходных данных в наглядной форме, минимальное число специальных требований, ориентация на русскоязычного пользователя. Он максимально облегчает работу с системой медицинским работникам, которые в массе своей являются неискушенными пользователями. Для ввода значений тех данных, которые представляют собой ограниченное множество, пользователю предоставляется возможность не набирать текст, а выбирать значение из заранее сформированного списка, который открывается при нажатии указателя списка в каждом окне.

Основным окном системы является экземпляр класса MainController. Внешний вид окна приведен на рис. 6. Основное окно управляет всей функциональностью системы. Это окно позволяет вызывать другие окна, предоставляющие доступ ко всем другим функциям системы. Основное окно содержит главное меню, которое разбито на пункты: «Поиск по имени пациента», «Поиск по основному диагнозу», «Поиск по названию операции». Подпункты меню «Поиск по имени пациента» предоставляют возможность пользователю системы найти в БД нужного пациента и показать его характеристики на экране монитора. Подпункты меню «Поиск по основному диагнозу» позволяют пользователю выбрать и визуализировать всех пациентов, которые имеют указанный диагноз. Подпункты меню «Поиск по названию операции» также находят всех пациентов, которым сделана указанная операция. Кнопка "Добавит пациента" переводит систему в режим ввода данных о новом пациенте.

Так как система разрабатывалась на объектно-ориентированном языке Java, то каждое окно представляет пограничным экземпляром класса, в котором запрограммирован внешний вид и

функциональность соответствующего окна пользовательского интерфейса.

С помощью предусмотренных запросов система формирует более 11 различных статистических отчетов по имеющимся данным о пациентах с различными заболеваниями толстой кишки и хирургической патологией органов брюшной полости, которые выводятся на печать или сохраняются в виде файлов. Кроме того, из главного диалогового окна есть возможность просматривать всю информацию о клиентках, занесённых в архив медицинских данных.

Заключение

Проблемы в области медицины, которые, как известно, не имеют однозначных решений, сопровождаются большим объёмом информации. Именно поэтому для получения решения здесь целесообразно проводить анализ и моделирование ситуаций с помощью современных компьютерных средств - специализированных информационных систем. В статье описывается разработка одной из таких систем (ИС АЛТК), предназначенной для анализа лечения заболеваний толстой кишки и органов брюшной полости.

В клинике общей хирургии ДонНМУ им. М. Горького на базе хирургического отделения №2 ДОКТМО за период с 2012 по 2015 гг. проведено оперативное лечение 986 пациентам с патологией толстой кишки, из них 128 (12,8%) пациентам произведены СОВ. Было установлено, что течение послеоперационного периода у пациентов, перенесших СОВ, существенно не отличалось от такового после выполнения стандартных вмешательств на толстой кишке. Однако, обсуждение результатов лечения показало, что для определения целесообразности и эффективности выполнения данного типа хирургического вмешательства необходимо более тщательная сравнительная оценка результатов выполняемых симультанных и стандартных оперативных вмешательств на основе достаточной большой статистической совокупности медицинских данных. Отсутствие единой информационной базы данных о результатах проведенных операций и автоматизированных средств интеллектуального анализа данных затрудняют объективную оценку СОВ и препятствуют их более широкому применению в клинической практике.

Поэтому в статье рассмотрен жизненный цикл проектирования информационной БД РЛТК в виде последовательности основных логических моделей на унифицированном языке моделирования UML, описывающие функции,

логическую структуру и схему реляционной базы данных информационной системы.

Программная реализация информационной системы выполнена на языке программирования Java в инструментальной среде IDE IntelliJ IDEA. Система, построенная по открытому принципу, предусматривает возможность расширения своих функций. Пользовательский интерфейс спроектирован с учётом удобства работы с системой. По запросам пользователя система автоматически формирует более 11 различных статистических отчётов по имеющимся данным о пациентах с различными заболеваниями толстой кишки и хирургической патологией органов брюшной полости.

В заключение хотелось отметить аспекты, в которых может использоваться ИС АЛТК:

Во-первых, система работает как электронный архив, накапливая в структурированном виде информацию о показаниях и результатах лечения больных с ЗТК и хирургической патологией органов брюшной полости, которым выполнены СОВ. Она обеспечивает надёжность сохранения и достоверность информации.

Во-вторых, система может использоваться для статистической обработки собранной информации о проведенных медицинских операциях в количественных и процентных значениях по разным параметрам и их комбинациям.

В-третьих, разработанный программный продукт может использоваться для формирования более рациональных вариантов хирургического лечения данной категории больных на основе использования БД РЛТК и внедрения методов интеллектуального анализа данных (технологии Data Mining).

Наконец, *в-четвёртых*, созданный вариант медицинской информационно-справочной системы в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений по лечению заболеваний желудочно-кишечного тракта.

Литература

1. Лебедев Г.С., Максаков В.В. Концепция портала «Медицинская наука» // Информационно-измерительные и управляющие системы. – 2014. - № 10, т.12. – С.1-6.
2. Евтихова Е.Ю., Кутырев Е.А., Гагуа А.К., Евтихов Р.М., Аль Бикай Рами А.А., Шурыгин С.Н. Возможности симультанных операций при послеоперационных вентральных грыжах// Альманах клинической медицины, Ярославль,- 2008, - Том 17.- С.120-121.
3. Баулина, Н. В. Симультанные эндохирургические операции в гинекологии и хирургии Текст. : Тезисы 8-го Всероссийского съезда по эндоскопической хирургии // Эндоскопическая хирургия. - 2005. - С. 14-15.
4. Аль Бикай Рами А.А., Познанский С.В., Кукушкин А.В. Результаты симультанных операций у пациентов с желчнокаменной болезнью // Вестник Ивановской медицинской академии, 2008. - Том13. - № 3-4,- С. 87-88.
5. Маховский В. З. Одномоментные сочетанные операции в неотложной и плановой хирургии Текст. / В. З. Маховский // Хирургия. - 2002. - № 6. - С. 46.
6. Евтихова Е.Ю., Шурыгин С.Н., Грачев И.С., Муха А.В., Лебедева О.А., Аль Бикай Рами А.А., Сорокин С. Ю. Симультанные операции при патологии органов брюшной полости и послеоперационных вентральных грыжах// Вестник трансплантологии и искусственных органов. - 2006. - № 3. - С. 44-45.
7. Аль Бикай Рами А.А., Познанский С.В., Кукушкин А.В. Результаты симультанных операций у пациентов с желчнокаменной болезнью // Вестник Ивановской медицинской академии, 2008. - Том13. - № 3-4,- С. 87-88.
8. Евтихова Е.Ю. Шумаков Д.В., Гагуа АХ, Аль Бикай Рами А.А., Лебедева О.А., Евтихов Р.М. Симультанные операции при послеоперационных вентральных грыжах // Вестник Ивановской медицинской академии, 2007. - Том 12. - № 1-2, - С. 66-68.
9. Кузин, М. М. Лапароскопическая и традиционная холецистэктомия: сравнение непосредственных результатов Текст. / М. М. Кузин [и др.] // Хирургия. - 2000. - № 2. - С. 25—27.
10. Мюллер Р.Д. Базы данных и UML: Проектирование. – М.: Лори, 2002.- 420 с.
11. Джозеф Шмюллер. Освой самостоятельно UML 2 за 24 часа. Практическое руководство. - М.: Вильямс, 2005. - 416 с.

12. Васвани В. MySQL: использование и администрирование. - М.: «Питер», 2011. - 368 с.

13. Стив Суэринг, Тим Конверс, Джойс Парк. PHP и MySQL. Библия программиста, 2-е издание. - М.: «Диалектика», 2010. - 912 с.

Борота А.В., Борота А.А., Кухто А.П., Базиян-Кухто Н.К., Федяев О.И., Бакаленко В.С. *Разработка информационной системы истории болезни органов брюшной полости пациентов.* В статье рассматривается разработка медицинской информационно-справочной системы, автоматизирующей процессы сбора, хранения, редактирования и предварительной статистической обработки данных для предоставления медицинским работникам необходимой информации о пред- и постоперационных периодах лечения болезней органов брюшной полости пациентов. Для решения данной задачи использовались современные методы и технологии построения информационных систем с базой данных реляционного типа. В результате построена объектно-ориентированная система открытого типа, которая в дальнейшем будет функционально расширяться до системы поддержки принятия врачебных решений с целью эффективного лечения заболеваний желудочно-кишечного тракта.

Ключевые слова: база данных, информационная система, UML-модели, медицина, органы брюшной полости.

Borota A.V., Borota A.A., Kukhto A.P., Baziyan-Kukhto N.K., Fedyaev O.I., Bakalenko V.S. *Development of information medical records system of the abdominal cavity of patients.* The article deals with the development of medical information and referral system that automates the collection, storage, editing and preliminary statistical data to provide medical professionals the necessary information about the pre- and postoperative treatment of diseases of the abdominal cavity of patients. To solve this problem, we used modern methods and technologies of information systems with relational database data type. As a result, based object-oriented open-end system that will continue to expand functional prior to medical decision support systems for the effective treatment of diseases of the gastrointestinal tract.

Keywords: database, information system, the UML-model, medicine, organs abdominal bowel disease

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 519.21

Принятие решений в системах с субъективной формой понимания интересов и представлений о ситуации выбора

Г.П. Виноградов, Н.Г. Виногорова
Тверской государственный технический университет
wgp272ng@mail.ru

Введение

Развитие идеи субъективно рационального выбора открыло возможность: 1) объяснить принятие решений субъектом в конкретных ситуациях; 2) предсказания принимающим решение возможных реакций другого субъекта в различных ситуациях; 3) решать задачу активного прогноза, когда управляющая сторона создает у управляемой стороны нужный образ будущего [8, 10]. Субъективно рациональный выбор предполагает, что его мотивация определяется как внешними, так и внутренними факторами. Внутренние факторы отражают интересы субъекта, индуцируемые его потребностями и этической системой, которой он придерживается. Оценки удовлетворенности текущей ситуацией целеустремленного состояния субъектом, как показано в [1], могут приводить к изменению структуры интересов субъекта и он ее может выбирать. Поскольку предпочтения субъекта в процессе выбора отражают его интересы, то можно определить множество G альтернативных вариантов структуры предпочтений, которые будем называть структурными альтернативами [2].

1 Концептуальные предположения

1. Выбор субъектом осуществляется на основе представлений о ситуации целеустремленного состояния.

2. Компоненты представления отражают различные аспекты понимания субъектом ситуации целеустремленного состояния и образуют информационную структуру представлений. Множество возможных вариантов представлений обозначим через X .

3. Для множества состояний окружения Ω множество наблюдаемых состояний окружения удовлетворяют условию $\Omega \cap X \neq \emptyset$, то есть представления субъекта могут содержать как объективную составляющую, так и фантомную.

4. Структурные альтернативы субъект выбирает в зависимости от оценок

удовлетворенности значениями свойств ситуации целеустремленного состояния.

5. Формирование представлений осуществляется на основе процедур восприятия, осознания и анализа согласно с когнитивными возможностями субъекта.

Активные интеллектуальные системы содержит в себе субъект интересов, объект интересов, средства реализации интересов и порождаемые ими отношения. В работах [1, 3] доказано, что формально интересы субъекта (в дальнейшем агента) могут быть представлены двумя показателями: удельная ценность ситуации целеустремленного состояния по результату и удельная ценность ситуации целеустремленного состояния по эффективности. Они позволяют оценить степень желательности для агента результатов и величину прикладываемых для этого усилий. С усилиями связано определение возможных способов действия, которые с формальной стороны следует рассматривать как управление. Оценку качества выбора следует связывать со стремлением к реализации интересов с некоторым «наилучшим результатом». Формально понятие «наилучший результат» следует характеризовать, например, отношением удельной ценности ситуации целеустремленного состояния по результату к удельной ценности ситуации целеустремленного состояния по эффективности [3]. Значения этого показателя являются значениями некоторой шкалы, характеризующей субъективное отношение агента к эволюции своего состояния в зависимости от выбранного управления, а также к эволюции системы в целом. Полярные значения этой шкалы: гибель – доминирование. Понятно, что оценки по этой шкале будут определять конкретное содержание интересов, их эмоциональное переживание, степень стремления к их реализации. На их значения будут оказывать влияние система ценностей и норм агента, а также этическая система. Если учитывать тот факт, что реализация интересов агента возможна только в рамках некоторой организационной системы, то можно считать, что интересы системы в целом и каждого агента

не противоположными, но, в общем случае, не совпадающими.

Как установлено в [4] система внутренних ценностей может рассматриваться априорно заданной и инвариантной лишь до тех пор, пока не возникнет ситуация возможной гибели агента или получение таких оценок свойств ситуации целеустремленного состояния, которые он не может считать удовлетворительными. С другой стороны систему ценностей следует рассматривать зависимой от принятой агентом этической системы, которая определяет для человека такие фундаментальные понятия, как «Добро» и «Зло» [6]. Этические системы, построенные на импликации этих понятий в терминах намерение (цель), действие, результат, определяются какими-то глубинными свойствами этноса, группы. Следовательно, понятие «с наилучшими результатами» зависит от этической системы через систему ценностей и норм, которая в свою очередь определяет структуру предпочтений на действиях агента. Отметим, что в нормативной теории принятия решений структура предпочтений агента считается априорно заданной [5].

В данной схеме формализации интересов агента можно определить его субъективные оценки желательности и возможности достижения различных значений удельной ценности ситуации целеустремленного состояния, и через них конкретные значения показателей качества «жизни», к которому стремится агент. Поскольку их значения формируются на основе субъективных оценок ценности и возможности реализации интересов, то интересы агента и соответствующие цели являются эндогенными, то есть формируемыми внутри системы. Структура и конкретные значения целей определяются степенью выражения в них интересов агента через субъективные оценки свойств компонент ситуации целеустремленного состояния (подробно об этом см. в [3]).

Стремление выжить, не допустить снижение удельной ценности ситуации целеустремленного состояния и обеспечить ее рост, данное в инстинктах, определяют второе свойство любой системы, проявляющей поведение. Оно связано с необходимостью непрерывного изучения своего окружения, а также совершенствования методов и способов действия. Его цель для агента состоит в поддержке прогрессивной эволюции себя как целостной системы.

2 Принцип формирования целей на основе интересов

Введем ряд утверждений:

1. Мотивацией поведения субъекта являются интересы и потребности. Характеризующие их показатели образуют иерархическую структуру, например, в смысле Маслоу.
2. Агент, как носитель интересов, выступает для объекта интересов в качестве управляющей системы.
3. Объект интересов агента может быть как пассивным, так и активным, обладающим свойством суверенности, активности, целеустремленности и интеллектуальности.
4. Субъект интересов может выбирать структуру предпочтений на множестве состояний объекта интересов, то есть его структура предпочтений не фиксирована.
5. Цель управления системой не задана экзогенно, а формируется внутри системы на основе согласования интересов и понимания обстановки.
6. Мотивация деятельности субъектов системы определяется их заинтересованностью в прогрессивной эволюции системы в целом и отдельных ее частей.

Согласно введенной аксиоматике мотивация на управление (действие) определяется заинтересованностью субъекта эволюцией объекта управления. Следовательно, существует набор переменных, которые агент наблюдает и воспринимает, и которыми он характеризует состояние объекта и свое окружение. Их восприятие и осознание является необходимым внешним условием, побуждающим субъекта к выбору управляющих воздействий, направленных на реализацию его интересов в ситуации выбора.

Заинтересованность агента в эволюции объекта интересов предполагает его отношение к состоянию окружения, которое порождается его представлениями о соответствии наблюдаемого состояния успешности реализации его интересов, намерений, интенций. Эти представления определяются многоуровневой системой обработки информации мозгом человека, структура которой с точки зрения постнеклассического принципа рациональности рассмотрена в [8]. Само отношение выражается на качественном уровне посредством вербальных оценок и касается всех компонент состояния, в том числе и самого субъекта выбора. Таким образом, представление порождает как отношение к состоянию, так и некоторое отношение к

возможности и эффективности с помощью управляющих воздействий реализовать интересы с «наилучшим результатом». Существование такого отношения позволяет субъекту ориентироваться при выборе управления (способа действия), а также определять содержательный смысл (кроме того и прагматический смысл) его стремления к реализации интересов с «наилучшим результатом».

Определение 1. Качественные характеристики, определяющие отношение субъекта к 1) состоянию окружения, 2) оценке направления его эволюции, 3) ценности, эффективности возможных результатов, 4) вероятности реализации интересов с точки зрения представлений о состоянии окружения, определяемое его внутренней системой интересов, ценностей и норм, а также его этической системой будем называть оценками ситуации целеустремленного состояния.

Поскольку представление о ситуации целеустремленного состояния строится на основе субъективных оценок состояния окружения и о себе самом, то можно утверждать, что выбор управляющих воздействий, направленных на реализацию интересов с наилучшим результатом» является **субъективно рациональным выбором**, осуществляемый на основе представлений о ситуации целеустремленного состояния. Состояние окружения в этом случае следует рассматривать как экзогенный фактор.

Ситуация целеустремленного состояния существует в сознании субъекта, отражает его индивидуальные особенности по моделированию состояния окружения, в частности представление о состоянии среды может быть фантомным. Поэтому следует ожидать отношения субъекта к своим представлениям или рефлексии своих представлений. Это факт обсуждался в работах [4, 7, 8]. В частности, предложено оценивать представления по критерию их полезности при реализации интересов. В любом случае следует постулировать факт использования человеком определенных процедур, повышающих его убежденность в полезности представлений. Следовательно, человек осуществляет выбор из множества возможных вариантов представлений, то есть он выбирает модель ситуации целеустремленного состояния в зависимости от степени убежденности в ее адекватности и полезности для перехода в желаемое состояние. Для этой цели он разрабатывает и реализует специальные процедуры, которые будем называть процедурами идентификации ситуации целеустремленного состояния. Цель их

применения – осуществление выбора модели ситуации целеустремленного состояния для получения оценок возможных вариантов результатов. Следовательно, ситуация целеустремленного состояния является эндогенным фактором в модели выбора субъекта и определяет его отношение к наблюдаемому состоянию, характеризуемое им с помощью набора качественных характеристик.

Эти рассуждения позволяют предложить следующую схему выбора управляющих воздействий:

1. Для каждого состояния окружения $\omega \in \Omega$ агент ставит в соответствие ситуацию целеустремленного состояния $x_\omega \in X$, $g \in G$, где множество G характеризует предпочтения на множестве X . Для ситуации целеустремленного состояния определяется ее удельная ценность E_φ , как оценка степени реализации интересов (ожиданий). Проводится оценка степени удовлетворенности ситуаций, а также оценка степени соответствия представлений о ситуации целеустремленного состояния состоянию Ω . В случае, если убежденность в адекватном отражении состояния Ω ниже некоторого порога, выполняется процедура идентификации ситуации.

2. Если удовлетворенность ситуаций целеустремленного состояния ниже заданного порога, то из множества G структурных альтернатив (оно соответствует позитивному опыту $G' \subseteq G$, когда с их помощью субъект наблюдал реализацию своих интересов) выбирается управляющее воздействие $c \in C$, позволяющее достичь желаемой ситуации целеустремленного состояния, ценность которой либо превышает пороговое значение, либо является оптимальной величиной при имеющихся возможностях. Определяется значение ценности желаемой ситуации целеустремленного состояния, строится дерево целей и способов действия, позволяющих ее достичь и конкретизирующее интересы субъекта. Переход к п.6.

3. В противном случае определяется возможность достичь приемлемого состояния с помощью других вариантов структурных альтернатив $g \in G$. Если это невозможно, то фиксируется возникновение проблемы, составляется проект ее преодоления путем соответствующих исследований и разработок. Их цель:

а) расширение множества способов действия ; б) расшивка ограничений и т.п. Для этой цели определяется направление эволюции в пространстве показателей и соответствующая возможностям величина шага.

4. Формируется план формирования нового намеченного способа действия $c \in C \uparrow$ и план расширения множества способов действия $C' \equiv C \uparrow$.

5. Определяется величина стимулирования для создания определенного уровня мотивации субъектов, задействованных в реализации интересов субъекта интересов.

6. Для ситуации целеустремленного состояния реализуются структурные управляющие воздействия в форме, определенной в п. 4-5.

Такую схему, вытекающую из аксиом, постулирующих мотивацию управления (целеустремленного поведения) будем называть схемой управления целенаправленной эволюции, определяемой стремлением субъекта к выживанию, сохранению достигнутого уровня или доминирования.

Согласно этой схеме субъект принимает управляющие решения, учитывая два типа условий: 1) экзогенных (объективных), порождаемых динамикой окружения и объекта интересов; 2) эндогенных (субъективных), порождаемых интересами субъекта.

Это возможно, если субъект обладает *стремлением к реализации интересов, к повышению вероятности их достижения за счет роста своего уровня знаний* закономерностях динамических процессов в окружении и динамики перехода объекта управления в различные состояния и обладает соответствующими *креативными способностями*. Варианты интересов могут быть представлены следующими терминами на оценках удельной ценности ситуации целеустремленного состояния: *выживание, сохранение достигнутого уровня жизни, возможность целенаправленного изменения среды, доминирование*.

Согласно аксиомам выбора управляющая альтернатива выбирается в зависимости от представлений о ситуации, которые формируются с помощью процедуры идентификации сообразно с состоянием. Тогда функция полезности должна быть определена на множествах способов действия, оценок ситуации $x \in X$ и состояний $\omega \in \Omega$. Пусть субъект имеет возможность выбирать структуру предпочтений из заданного множества структурных альтернатив G для получения приемлемых оценок удовлетворенности ситуации целеустремленного состояния. Тогда его функция полезности будет зависеть от структурной альтернативы $g \in G$ как от параметра вида $Efg : (C \times \Omega \times X) \rightarrow R^1$, представляющая априорные предпочтения на

управляющих альтернативах $c \in C$ в соответствии с условием:

$$c' \succ c \Leftrightarrow Efg(c', \omega, x) > Efg(c, \omega, x).$$

Подробно содержательный смысл и способы задания функции полезности для агентов с субъективно рациональной формой поведения рассмотрены в [3].

Как показано в [3] функция полезности

$Efg(c', \omega, x)$ для агента имеет смысл оценки удельной ценности ситуации целеустремленного состояния и представляет его априорные внутренние предпочтения на управляющих альтернативах в зависимости от состояний и ситуаций, а также его системы ценностей и норм, задаваемых этической системой. Так как выбор осуществляется в условиях неопределенности, то это предполагает в ведение в модель выбора предположений и правил устранения неопределенности, которые еще называют гипотезами детерминизма. Различные варианты элиминации неопределенности рассмотрены в работах Д.А. Новикова [9]. Здесь же ограничимся замечанием, что правило управления естественно выбирать из условия достижения максимума ожидаемой удельной ценности ситуации целеустремленного состояния.

Поскольку понятие ситуации целеустремленного состояния существует только в сознании агента, то суть ее формирования состоит в выборе модели ситуации целеустремленного состояния в зависимости от состояния окружения. Очевидно, что если возможные модели рассматривать в роли альтернатив идентификации состояния окружения, то требуется некоторый критерий оценки их полезности и качества. Поскольку модель ситуации целеустремленного состояния является основой для выбора, то функция полезности может служить основанием для определения смысла и структуры требуемого критерия. Действительно, пусть для каждого состояния $\omega \in \Omega$ определена пара переменных $(x^*, c^*)_s \in X \times C$, на которой функция полезности достигает своего максимального значения. Пусть теперь из некоторых соображений в состоянии $\omega \in \Omega$ выбрана ситуация $x \in X$ в роли результата идентификации, и с учетом этого выбрано управляющее воздействие $c \in C$. В таком случае будут иметь место определенные «потери полезности» в сравнении с возможным наилучшим выбором.

Поскольку, в общем случае, окружение субъекта интересов является неопределенным и

слабо формализуемым и наблюдаемая траектория состояний зависит от состояния окружения и от правила выбора управляющих, действий, то качество представлений будет зависеть от используемых процедур идентификации состояния, которые можно рассматривать как средство детерминации представлений. Однако получить оценки потерь в этом случае возможно только по результатам реализации выбора. Поэтому человек на значениях возможных значений функции полезности в ситуации неопределенности строит два типа оценок: первый связан с оценками полезностью представлений для получения желаемых состояний, а второй – с оценками возможности их достижения в виде оценок степени сомнения в правильности (адекватности) представлений.

Определим степень сомнения в правильности представлений о ситуации выбора в состоянии окружения $\omega \in \Omega$ относительно возможности достижения желаемого состояния $o \in O$ располагаемыми способами действия $c \in C$ как степень стремления у агента получить дополнительную информацию о количестве и степени влияния факторов окружения для повышения своей информированности $X \uparrow$.

Поскольку степень стремления к получению дополнительной информации изменяется от нуля до некоторого максимума, то эти оценки можно рассматривать как значения шкалы лингвистической переменной степень сомнения.

Оценки полезности представлений связаны с ценностью появления желаемых состояний $o \in O$ в окружении $\omega \in \Omega$ при выборе способа действия $c \in C$ на основе представления $x \in X$, которые также являются значениями лингвистической переменной с тем же названием.

Очевидно, что интегральная оценка потерь полезности может быть получена по известным правилам теории нечетких множеств. Соответствующую оценку, как и в теории статистических решений, будем называть «риском»

Таким образом, в схеме целеустремленного управления для описания качества управления естественно использовать критерий, имеющий смысл ожидаемой удельной ценности ситуации целеустремленного состояния, а для описания качества идентификации состояния окружения – критерий, имеющий смысл лингвистической переменной риск. Ясно, что выбираемые таким образом правила будут определенным способом взаимозависимы. В этих условиях проблема выбора имеет игровое содержание, а ее «наилучшее» решение состоит в построении

некоторого устойчивого компромисса между достижением максимальной ожидаемой полезности и минимального риска. Такой компромисс принято называть «равновесием» [9]. Определение подобного равновесия естественно рассматривать в качестве внутренней цели управления, обеспечивающей требуемое значение по шкале удовлетворенности.

С учетом выполненных рассуждений окончательно концепцию управления в условиях принятой аксиоматики определим следующими основными положениями.

1. Наблюдение и обнаружение состояния является необходимым, но не достаточным условием осуществимости выбора управления.
2. Достаточное условие осуществимости выбора управления определяется заданием отношения субъекта к состоянию, определяемого некоторой качественной характеристикой, называемой ситуацией целеустремленного состояния.
3. Вследствие того, что причинно-следственные связи, определяющие поведение окружения, недоступны непосредственному наблюдению, возникает необходимость выполнения процедур идентификации состояния окружения, их цель – выбор модели ситуации наблюдаемого состояния.
4. Выбор правила управления выполняется по критерию ожидаемой полезности.
5. Выбор правила идентификации выполняется по критерию риска.
6. Проблема выбора правил управления и идентификации имеет игровое содержание, «наилучшее» решение которой состоит в построении устойчивого компромисса, называемого «равновесием».
7. Построение и использование равновесных правил управления и идентификации является внутренней целью управления.

Сформулированные положения определяют концепцию «целеустремленного управления активными системами с эндогенно формируемыми целями».

Согласно введенной аксиоматике объект интересов может быть активным, динамическим при этом природа его динамики не связана с механическим движением. Если его эволюция описывается правилом $f : X \times C \rightarrow X$ и f – нечеткая функция вследствие использования лингвистических

оценок, то соответствующая система является нечеткой системой, состояние которой в момент $t+1$ есть условное по x_t и c_t нечеткое множество, характеризуемое функцией принадлежности вида $\mu(x_{t+1} | x_t, c_t)$. В условиях естественно полагать, что эволюция объекта описывается нечетким марковским процессом.

Без ограничения общности можно полагать, что множество состояний окружения Ω некоторым способом упорядочено и их значения случайны, то естественно полагать, что задано априорное распределение вероятностей $\beta(\Omega)$ начальных состояний.

Согласно концепции целеустремленного управления выбор управляющих альтернатив осуществляется в ситуации целеустремленного состояния, которая является качественной характеристикой, определяющей отношение субъекта к состоянию. Возможное множество ситуаций X может быть лишь конечным и при этом должно выполняться условие $|X| \leq |\Omega|$. Следовательно, к определенным выше двум аспектам интересов концепция целеустремленного управления вводит третий аспект, который связан с необходимостью идентификации ситуации в зависимости от наблюдаемого состояния.

Естественно полагать, что для каждого аспекта интересов существуют альтернативные варианты их реализации, и заданы соответствующие множества допустимых альтернатив. Тогда, следуя концепции целеустремленного управления, положим, что для аспекта интересов, связанного с управлением эволюцией состояний, задано множество C способов действия – управляющих альтернатив. Поскольку управляющие альтернативы выбираются в зависимости от ситуаций $x \in X$, то естественно полагать, что существуют ограничения на допустимость управляющих альтернатив в зависимости от ситуации $x \in X$. Такие ограничения естественно задавать включениями вида $C_x \subset C, x \in X$, либо $C_x \subseteq C, x \in X$.

Как показано выше, чтобы ситуация целеустремленного состояния была бы доступна сознанию агента, необходимо выполнение ее идентификации, содержание которой сводится к выбору ситуации из заданного множества в зависимости от состояния, то есть $x \in X$. Очевидно, альтернативы идентификации зависят от состояния $s \in S$, то есть $X_s \subset X, s \in S$.

Субъект заинтересован в реализации своих интересов для этого он выбирает структуру своих предпочтений $g \in G$ из

заданного множества G структурных альтернатив.

Структура предпочтений отражает только интересы агента. Поэтому их выбор не может зависеть от состояний и ситуаций. То есть структурные альтернативы должны рассматриваться в качестве общего для состояний и ситуаций параметра, определяющего закономерность динамики состояний, как объекта интересов, так и окружения. Естественно полагать, что в составе активной системы субъект может наблюдать состояния и принимать управляющие решения лишь в дискретные моменты времени. С учетом этого процесс управления будем понимать как процесс последовательного принятия решений с дискретным временем.

Если субъект не меняет на интервале $n > 1$ стремления достичь определенного состояния и не меняет структуру своих предпочтений, то такая структурная альтернатива является стратегической, в противном случае – тактической.

Согласно введенному выше предположению эволюция объекта интересов описывается нечетким марковским процессом в пространстве состояний S . Применение управляющих воздействий в дискретные моменты времени порождает управляемый марковский процесс с дискретным временем с тем же множеством состояний S . Динамика такого процесса определяется переходной функцией, задающей вероятности одношаговых переходов на множестве состояний в зависимости от выбора управляющих альтернатив $u \in U$. При этом она будет зависеть также и от структурной альтернативы $g \in G$, но как от параметра. Обозначим переходную функцию управляемого процесса символом $qg(S | S \times C)[2]$.

Согласно концепции целеустремленного управления должна быть задана функция полезности $Efg : (C \times S \times X) \rightarrow R^1$, представляющая априорные предпочтения на управляющих альтернативах $c \in C$ в соответствии с условием:

$$c' \succ c \Leftrightarrow Efg(c', s, x) > Efg(c, s, x).$$

Введенные предположения определяют условия принятия управляющих решений, где функция полезности зависит от структурной альтернативы $g \in G$ как от параметра, а управление выполняется в соответствии со схемой ситуационного управления.

3 Требования к информационной структуре представлений

В соответствии с введенными положениями, используя результаты работы [2], определим соответствующие объектам интересов носители информации как набор следующих формальных объектов:

$$I = \{S, \beta_0(S), X, [X_s \subset X, s \in S], C, [C_x \subseteq C, x \in X],$$

$$G, q^G(S | S \times C), E^G(C \times S \times X), g \in G\},$$

где S – множество состояний; $\beta(S)$ – априорное распределение на множестве состояний; X – множество ситуаций; $X_s \subset X$ – ограничения на допустимость ситуаций в роли альтернатив идентификации в зависимости от состояний $s \in S$; C – множество управляющих альтернатив; $C_x \subseteq C$ – ограничения на допустимость управляющих альтернатив в зависимости от ситуаций $x \in X$; G – множество структурных альтернатив; $q^G(S | S \times C)$ – переходная функция из $S \times C$ в S ; $E^G(C \times S \times X)$ – функция полезности, представляющая априорные предпочтения на альтернативах $c \in C$ в зависимости от состояний $s \in S$, ситуаций $x \in X$ и структурных альтернатив $g \in G$.

Особенность информационной структуры I заключается в том, что выбор управляющих воздействий зависит от представлений о ситуации выбора, формируемых на основе процедур идентификации. В этих условиях закономерность динамики ситуаций априори не может быть задана. Поэтому агент использует субъективные оценки закономерности динамики состояний, определяемой переходной функцией $q^G(S | S \times C)$ из $S \times C$ в S .

Если субъект сомневается в своих представлениях об элементах информационной структуры, то он использует дополнительную информацию для формулирования некоторого правдоподобного приближения. Например, сформировать некоторое множество гипотез Γ .

Пусть, например, в базовой информационной структуре I переходная функция $q^G(S | S \times C)$ не задана, но задано множество гипотез Γ о ней. Тогда формально можно полагать, что переходная функция $q(g, \gamma)(S | S \times C)$ зависит от некоторого параметра γ принимающего значения из заданного множества Γ , но истинное значение такого параметра остается неизвестным. В этих условиях информационная структура будет определяться набором объектов вида

$$I\Gamma = \{S, \beta(S), X, [X_s \subset X, s \in S], Y, [Y_x \subseteq Y, x \in X], G, \Gamma, q(g, \gamma)(S | S \times X),$$

$$E^G(Y \times (S \times X)), g \in G, \gamma \in \Gamma\}$$

Очевидно, что это вместе с выбором альтернатив потребует также выбора в определенном смысле «наилучшей» гипотезы о переходной функции.

4 Игровая модель выбора

Введенные предположения фиксируют существование двух аспектов интересов субъекта, один из которых определяется заинтересованностью в эволюции объекта интересов, а другой – в выборе структуры интересов. Концепция целеустремленного управления порождает третий аспект интересов, связанный с необходимостью идентификации ситуации в зависимости от наблюдаемого состояния. В соответствии с этими тремя аспектами информационная структура I предполагает задание множества C управляющих альтернатив, множества G структурных альтернатив и множества X альтернатив идентификации. Предполагается задание также функции полезности $E^G(C \times S \times X)$ и переходной функции $q^G(S | S \times C)$ из $S \times C$ в S . Это позволяет задать критерий качества выбора управления, имеющего смысл ожидаемой полезности, и критерий качества выбора правила идентификации, имеющего смысл риска. Для выбора структурных альтернатив может быть введен критерий качества, например, отражающий степень удовлетворения интересов.

Таким образом, при принятии решения о способе действия субъект для удовлетворения своих интересов использует три множества альтернатив. Выбор альтернативы из соответствующего множества производится по своему индивидуальному критерию качества и его естественно связать с некоторой виртуальной оперирующей стороной, имеющей в пределах своей компетенции свободу выбора. Интересы субъекта являются доминирующими для выделенных сторон, и он выступает для них в виде центра управления. Поскольку интересы всех игроков взаимозависимы, то такая игра относится к классу игр с корпоративными интересами. Ее решение возможно на множестве согласованных компромиссных вариантов. Оно будет устойчивым, если его невозможно улучшить без ухудшения хотя бы одного из критериев. Подобное моделирование выбора будет возможно, если для структуры представлений определить наборы носителей

информации для построения требуемых критериев. Это позволяет гарантировать, что для каждой выделенной стороны задача изолированного выбора ее «наилучшей» индивидуальной альтернативы по соответствующему критерию будет разрешима. Тогда задача совместного отыскания компромисса, удовлетворяющего требованиям корпоративной устойчивости, также будет разрешима.

Заключение

В статье представлена концептуальная схема моделирования принятия субъективно рациональных решений, предназначенная для понимания специфики управления эволюцией инновационных систем. В этом контексте важна проблема сборки субъектов, идентифицирующих себя с системой, в которой происходит их деятельность, и регулирующих свою активность с учетом ее влияния на результаты этой системы. Принято разделить процесс принятия решений на этапы: 1) формирования представлений о свойствах ситуации выбора; 2) собственно принятие решения о способе действия; 3) построение алгоритма (плана) реализации решения; 4) реализация алгоритма; 5) оценка результатов. С формальной точки зрения рассмотрены два первых этапа. Качество принятого решения предложено оценивать по критерию удельной ценности ситуации целеустремленного состояния по результату, качество формирования представлений о ситуации выбора – критерием потерь ценности. Показано, что в этом случае принятие решений следует рассматривать как корпоративную игру с противоположными интересами.

Литература

1. Виноградов, Г.П. Моделирование поведения агента с учетом субъективных представлений о ситуации выбора / Г.П. Виноградов, В.Н. Кузнецов // Искусственный интеллект и принятие решений. 2011. № 3. с. 58-72.
2. Баранов В.В. Динамические равновесия в задачах стохастического

управления и принятия решений при неопределенностях // Изв. РАН. Теория и системы управления. 2002. №3. С. 77-93.

3. Виноградов, Г.П. Методы и алгоритмы принятия решений в автоматизированных системах управления производствами с непрерывной технологией на основе субъективных представлений: монография. / Г.П. Виноградов. Тверь: ТГТУ, 2013. 256 с.

4. Виноградов, Г.П. Формирование представлений агента о предметной области в ситуации выбора/ Г.П. Виноградов, Г.П. Шматов, Д.А. Борзов // Программные продукты и системы. №2 (110), 2015. – С. 83–94.

5. Ларичев О.И., Мовшович Е.М. Качественные методы принятия решений. Вербальный анализ решений. – М.: Наука, 1996.

6. Lefebvre V. Algebra of Conscience. Dordrecht/Boston/London.: Kluwer Academic Publ. – 2001.

7. Виноградов Г.П., Кузнецов В.Н. Целеуказание на основе представлений агента о целеустремленной ситуации в слабоструктурированной среде. XII Всероссийское совещание по проблемам управления ВСПУ–2014. Москва, 16-19 июня 2014г.: Труды [Электронный ресурс]. М.: Институт проблем управления им. В.А. Трапезникова РАН, 2014. Электронные текстовые данные. ISBN 978-5-91450-151-5. Номер государственной регистрации 0321401153. – с. 7907-7918.

8. Тоценко В.Г. Методы и системы поддержки принятия решений. – Киев: Наукова думка, 2002.

9. Новиков Д.А., Иващенко А.А. Модели и методы организационного управления инновационным развитием фирмы. – М.: ЛЕНАНД, 2006. – 336 с.

10. Лепский В.Е. Рефлексивно-активные среды инновационного развития. М.: «Когито-Центр», 2010. – 255 с.

Виноградов Г.П., Виноградова Н.Г. Принятие решений в системах с субъективной формой понимания интересов и представлений о ситуации выбора. Эффективность рефлексивного управления поведением людей в организационно-технологических системах особенно в условиях неопределенности и риска зависит от представлений у управляющего органа о том, как люди выявляют и воспринимают объективные признаки складывающихся ситуаций, строят оценки, вырабатывают и согласуют формы поведения. **Цель работы:** разработка концептуальной схемы принятия решений, учитывающей процессы восприятия, осознаний и понимания свойств ситуации выбора для получения возможности: объяснения принятия решений субъектом в конкретных ситуациях; предсказания принимающим решение возможных реакций другого субъекта в различных ситуациях; решения задачи создания у управляемой стороны нужного образа ситуации выбора. **Результаты:** представлена концептуальная схема моделирования принятия субъективно рациональных решений, предназначенная для понимания специфики управления эволюцией инновационных систем, в которых особую значимость приобретает гармоничность взаимодействия «субъект-проект-среда». В этом контексте важна проблема сборки субъектов, идентифицирующих себя с

системой, в которой происходит их деятельность, и регулирующих свою активность с учетом ее влияния на результаты этой системы. Определено, что мотивация активности субъекта связана с интересами и со стремлением к их реализации. Для формального описания интересов субъекта введены два показателя: удельная ценность ситуации целеустремленного состояния по результату и удельная ценность ситуации целеустремленного состояния по эффективности. Предложено разделить процесс принятия решений на этапы: 1) формирование представлений о свойствах ситуации выбора; 2) собственно принятие решения о способе действия; 3) построение алгоритма (плана) реализации решения; 4) реализация алгоритма; 5) оценка результатов. С формальной точки зрения рассмотрены два первых этапа. Качество принятого решения предложено оценивать по критерию удельной ценности ситуации целеустремленного состояния по результату, качество формирования представлений о ситуации выбора – критерием потерь ценности. Показано, что при принятии решения о способе действия субъект для удовлетворения своих интересов использует три множества альтернатив: множество способов действия, множество вариантов представлений, множество стратегий идентификации. Выбор альтернативы из соответствующего множества производится по своему индивидуальному критерию качества. Предложено его связать с некоторой виртуальной оперирующей стороной, имеющей в пределах своей компетенции свободу выбора. Интересы субъекта являются доминирующими для выделенных сторон, и он выступает для них в виде центра управления. Показано, что в этом случае принятие решений следует рассматривать как корпоративную игру с не противоположными интересами.

Ключевые слова: принятие решений, моделирование выбора, теория игр, согласование представлений, активные системы, идентификация.

Vinogradov G.P., Vinogradova N.G. Decision making in systems with the subjective form of understanding of interests and ideas about a situation of choice. Due to the fact that the effectiveness of reflexive control the behavior of people in organizational and technological systems especially in the conditions of uncertainty and risk depends on the views of the governing body about how people identify and perceive objective evidence of factual situations, build assessments, produce and agree on forms of behavior. **Goal:** to develop a conceptual framework of decision-making that takes into account the processes of perception, awareness and understanding of the properties of choice situations to be able to: explain the decision-making entity in specific situations; prediction decision possible reactions of another subject in different situations; the creation of a managed side the desirable image of the situation of choice. **Research methods:** General methodology and methods of system analysis, analytical modeling, the work also uses methods theories: sets of active systems, decision making, artificial intelligence, fuzzy systems and fuzzy logic, game theory. **Results:** the conceptual diagram of the simulation of the adoption of the subjectively rational decisions designed to understand the specificity of management evolution of innovation systems in which of particular importance is the harmonious interaction of the "subject-project environment. In this context, an important problem of the Assembly of constituent entities identifying themselves with a system where their activities and regulating their activity, given its impact on the results of this system. It is determined that motivation of activity of the subject connected with the interests and with the desire to implement them. For a formal description of the interests of the subject introduced two indicators: the specific value of a purposeful state situation, the result and the specific value of a purposeful state situation. It is proposed to divide the decision process into stages: 1) formation of ideas about the properties of choice situations; 2) the actual decision-making about mode of action; 3) build an algorithm (plan) implement the solution; 4) implementation; 5) evaluation of results. From a formal point of view, are considered the first two stages. The quality of the decision proposed to be assessed against the specific value of a purposeful state situation according to the result, the quality of formation of representations about a situation of choice – the criterion of loss of value. It is shown that when deciding on the mode of action of the subject to suit their interests uses three sets of alternatives: multiple ways of action, lots of view options a lot of strategies of identification. The choice of an alternative from the set is your personal criterion. It is proposed to associate with a virtual operating party that has within its competence the freedom of choice. The interests of the subject are dominant for the selected parties, and he speaks to them in the form of control center. It is shown that in this case decisions should be considered as a corporate game with not opposite interests. Endogenous aim of the subject of the decision in this case is this distribution of effort between stages, to achieve the maximum satisfaction from the desired state. Since the interests of all players are interdependent, then its solution is possible on many of the agreed compromise.

Keywords: decision making, choice modeling, game theory, coordination of ideas, active systems, identification.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.942

Структура модели для прогнозирования поведения групп лиц

С.Ю. Землянская, А.В. Сложеницын, Н.К. Андриевская
ГОУВПО «Донецкий национальный технический университет»
SlogenitsinAV@gmail.com, zsaa@ya.ru, nataandr@yandex.ru

Землянская С.Ю., Сложеницын А.В., Андриевская Н.К. Структура модели для прогнозирования поведения групп лиц. Рассмотрено общее состояние разработок в области моделирования поведения человека и поведения сообществ. Выявлены достоинства и недостатки существующих моделей и готовых программных комплексов, с точки зрения прогнозирования поведения групп лиц. Определены основные критерии, используемые для построения новой модели поведения человека и прогнозирования поведения групп лиц, лишенной недостатков проанализированных продуктов. Выбраны методы и подходы для создания новой модели, описана ее структура.

Ключевые слова: моделирование поведения, прогнозирование деятельности, искусственный интеллект, агентное моделирование, искусственная нейронная сеть.

Введение

На сегодняшний день человечество достаточно полно изучило поведение индивидуума и поведение толпы, выявило множество факторов, определяющих их тип поведения и причинно-следственные связи, приводящие к различным событиям. Однако к настоящему времени программных реализаций моделирования поведения человека немного.

Для решения задач моделирования или прогнозирования поведения применяются различные методы: соцопросы, сбор информации о предпочтениях человека при его обращении к поисковым системам в интернете, статистический анализ прошлых реакций человеческих масс и прочие. Но все эти методы зачастую оказываются недостаточными для качественного прогнозирования поведения индивидуума.

Многие компании, страны, области науки заинтересованы в получении ответов о реакции и действиях различных людей в ответ на всевозможные события: выпуск новой продукции, проведение митинга, чрезвычайное происшествие и прочее.

Весьма важным направлением моделирования человеческой деятельности является моделирование преступной деятельности и прогнозирование криминогенной обстановки в городе. При успешных результатах построения модели станет возможным прогнозирование широкого спектра человеческой деятельности.

В данной статье основной целью исследования является обзор текущих разработок для моделирования поведения человека, а также выделение на их основе методов и подходов, наиболее подходящих для программной реализации системы прогнозирования преступной деятельности групп лиц.

Основные подходы к моделированию поведения человека

Так что же такое моделирование поведения человека? В самом общем смысле – это возможность предсказать человеческие реакции и действия в ответ на любые события. Но не стоит забывать, что поведение человека состоит из множества факторов: психология субъекта, его социальные связи, его прошлый опыт и т.д. Учитывая количество внешних и внутренних факторов, которые влияют на поведение человека, а также то, что сам человек не может точно сказать, почему он поступил так, а не иначе, создание модели поведения – отнюдь нетривиальная задача.

Сегодня моделирование поведения человека – это многоотраслевое междисциплинарное направление. В основном, моделирование поведения человека и групп лиц разделяют по задачам моделирования, например: моделирование поведения толпы, моделирование поведения отдельного индивида в самоорганизующейся толпе, моделирование деятельности единого субъекта, моделирование влияния поведения отдельных индивидов на изменение общего состояния группы лиц и т.д.

В первую очередь, различные задачи подразумевают разные подходы и средства для моделирования. Так, для моделирования поведения групп лиц чаще используют методы роевого интеллекта, элементы теории игр, аппарат искусственных нейронных сетей и прочие. В то же время для моделирования поведения индивида в группе используются методы агентного моделирования, математический аппарат нечеткой логики, некоторые концепции модели жизнеспособной системы.

Примером использования перечисленных технологий, может выступать модель движения толпы на основе роевого поведения насекомых, исследуемая Карповым В.Э. в статье «Коллективное поведение роботов. Желаемое и действительное» [1]. Другим примером может служить эволюция агентов

и их поведение в различных средах, рассматриваемые Петрушаном М.В. и Самариним А.М. в работе «Эволюционная модель поведения агентов в среде, побуждающей к коллективным действиям» [2].

Естественно, наиболее прогрессивные результаты получаются при соединении нескольких подходов, но хотя совмещение моделирования поведения толпы и поведения отдельного индивида довольно перспективно, оно сопряжено с рядом проблем. В частности: большой объем обрабатываемых системой данных; проблема качественного создания математического аппарата, способного корректно учитывать «сложного» индивида в толпе; ресурсоемкий процесс создания (в том числе, не всегда экономически обоснованный), по сравнению с отдельным моделированием поведения толпы и индивида, для более узких, конкретных задач.

Обзор разработок в области моделирования поведения человеческих сообществ

Многие исследовательские центры и отдельные фирмы уже создали ряд продуктов, позволяющих в определенной степени прогнозировать поведение групп лиц. Рассмотрим некоторые из них.

Программный продукт DI Guy – модуль искусственного интеллекта, моделирование поведения толпы с учетом каждого индивида [3].

Программа разработана и представлена компанией VT MAK. Основное ядро системы для моделирования поведения человека представлено в базовом пакете DI Guy. Дополнительными элементами являются DI Scenario, DI SDK, DI Motion editor. С их помощью возможно разрабатывать новые сценарии поведения и ситуативные среды, добиваться более реалистичного поведения отдельного человека (вроде движения ногами или хождения по кругу в ожидании); добавлять отображение лицевой анимации в формате 3D.

Внедренные в программу искусственный интеллект и графические возможности позволяют создавать реалистичные уличные драки, митинги, военные операции, допрос подозреваемых и многие другие ситуации. В связи с хорошо проработанными возможностями для моделирования поведения людей во время военных действий, большой интерес к данному проекту проявляют американские военные. Причем здесь важно заметить: при моделировании столкновения в мирном районе города с вражескими солдатами можно моделировать поведение не только этих солдат, но и поведение мирных жителей (каждого индивидуально), а также различных животных (кур, коров, лошадей). При качественном моделировании открывается возможность обучения солдат определять вероятных противников по жестам и мимике, даже с учетом поведения людей другой культуры [4].

Поскольку это коммерческий продукт,

компания неохотно раскрывает использованные для моделирования технологии. Известно, что проект опирается на агентное моделирование, в котором отдельный человек – это агент с набором характеристик, реагирующий на деятельность других агентов (людей). Однако детали реализации агентов и их взаимодействия не раскрываются.

Плюсами данной системы является возможность моделирования большого количества сцен с сильной детализацией, а также возможность моделирования в этих сценах различных построек, транспортных средств, животных. Минусом является заданная и мало изменяемая модель поведения человека – основные реакции заданы как «реакция на агрессию» в различных проявлениях. Также имеется ряд характеров для разных людей, но в них не учитываются особенности прошлого человека или его социальные связи.

Институт MOVES при военно-морской высшей школе США в Монтерее, штат Калифорния, занимается моделированием человеческого поведения [5]. Так как это военная школа, то основное направление моделирования – военные операции и реакция людей на них. В данной области, разработчиками были достигнуты хорошие результаты. Однако, дополнительная разработка направлена еще и на социальное и культурное моделирование. Из таких разработок данного института хотелось бы упомянуть проект FOCUS под руководством Стива Холла и Джеффа Апплегета, а также когнитивное и перцептронное моделирование. Первый проект позволяет строить модель социальной сети населения, которая адекватно реагирует на работу СМИ и помогает определить тип телерадиовещания, при котором население будет настроено максимально лояльно к армии. Эта разработка поможет создать программу профилактики как для снижения числа терактов в оккупированном городе, так и для снижения уровня преступности. Дополнительно разработчики планируют создать функционал для прогнозирования наиболее неблагонадежных (опасных) мест. Второй проект создается для улучшения понимания, как информация, доступная отдельному лицу, может быть интегрирована в подобие человеческого сознания для прогнозирования последствий различных направлений деятельности. То есть программа сможет давать ответ на вопрос, как человек воспринимает определенную деятельность и о возможных действиях человека, в ответ на какие-либо действия. [6]

В данном продукте в качестве основы для моделирования боя используется модель искусственного интеллекта, описывающая поведение объектов с точки зрения иерархических сетей. В звеньях этих сетей расположены вероятностные варианты поведения для различных ситуаций.

В процессе моделирования работы СМИ используется принцип географического положения, с помощью которого определяется, какие слои населения имеют доступ к системам

телерадиовещания. Затем определяется зона распространения информации «от человека к человеку», после чего моделируется реакция на вещание заданных объектов из заранее сформированного перечня.

Процесс создания проекта «Когнитивное и перцептронное моделирование» все еще находится на стадии разработки. В связи с этим информации о построенной модели немного. Единственное, что достоверно известно – текущие исследования основаны на многослойной нейронной сети, обучающейся без учителя.

Работа данного института позволяет понять, насколько различны технологии, применяемые для разных задач моделирования поведения. Это и является основным минусом каждого из представленных институтом проектов. Но, несмотря на это, каждый из проектов дает положительные результаты моделирования в своей области.

Наиболее подходящий для прогнозирования преступной деятельности проект – это моделирование боя на основе иерархических сетей. Однако использование данной модели становится практически невозможным при преступности менее 90-95% от общего числа исследуемых объектов. Это связано с тем, что на поле боя практически все лица являются прямыми участниками событий и обязаны реагировать в соответствии с общей тенденцией поведения. Данное обстоятельство практически исключает этот подход в области прогнозирования преступлений.

Еще одним интересным проектом стал **проект Марко Страно и Роберта Брузоне** [7]. Они собрали команду из психологов, медицинских экспертов и детективов. Целью этой команды стала разработка статистической модели, которая связывает профиль преступника с его / ее действиями, решениями, а также относительными последствиями на месте преступления.

Марко Страно и Роберта Брузоне определили переменные, которые играют существенную роль при характеристике поведения преступника во время насильственного преступления. Сто пятьдесят из этих переменных были выбраны путем анализа литературы и путем опроса выборки из нескольких сотен детективов и медицинских экспертов из разных стран, которые обычно участвуют в расследовании убийства. Эти переменные они используют для построения так называемых сетей Байеса. Байесовские сети являются графическими моделями, которые извлекают знание данной системы из эмпирических данных и отображают причинно-следственные связи между всеми соответствующими переменными. Используя условные вероятности, они могут определять степень, в которой переменные могут влиять друг на друга, даже если основные механизмы деятельности неизвестны. Данная работа направлена на определение профиля преступника и моделирование его возможного поведения. [7]

Плюсом этой системы можно считать хорошие

показатели моделирования. Это объясняется тем, что переменные для системы собирались много лет из статистически верных данных. Минусами же являются: сложность разработки модели Байесовской сети такого масштаба и большие вычислительные мощности, требуемые для однократного моделирования. Так, для каждой переменной существует от 2 до 30 возможных значений, и многие переменные связаны с пятью и более другими переменными, при том, что переменных уже сейчас насчитывается 150 [7, с 5].

Инструмент имитационного моделирования AnyLogic был разработан российской компанией The AnyLogic Company. Данная программная среда создана с использованием многоподходного моделирования. Объединение системной динамики, агентного и дискретно-событийного моделирования позволяет использовать пакет программ AnyLogic в разнообразных областях знаний. [8]

Проект включает в себя различные библиотеки, написанные на языке Java. Эти библиотеки реализуют отдельные возможности системы, такие как моделирование физической окружающей среды, имитацию операций сортировочной станции, интерактивную анимацию 2D и 3D моделей, моделирование движения автомобилей и многое другое.

Применение агентного моделирования позволяет использовать AnyLogic для моделирования поведения человеческих масс. На текущий момент уже есть некоторые наработки в этой области, например, «Моделирование поведения толпы» [9].

Минусами этого продукта является отсутствие возможности добавления большого количества характеристик агенту, что не позволяет в полной мере имитировать деятельность человека, а также строго заданные отношения между агентами, что не дает возможности организовать социальную связь, близкую к человеческой.

Некоторые перспективные разработки все еще не готовы, но их разработчики ожидают получить хороший результат. Примером такой разработки является **проект «Модель поведения человека»** – это неформальное сообщество исследователей, занимающихся в той или иной форме компьютерным моделированием человеческого поведения [10].

Проект представляет собой модульную программу с возможностью включения или отключения основных компонентов. Компонентами являются «Эмоциональный модуль», «Модуль опыта», «Модуль внешней среды», «Модуль энергетического баланса», «Модуль привыкания и автоматизмов» и прочие. Центральный элемент модели – человек. Вторым по важности элементом модели является деятельность человека, то есть его конкретные действия. На текущий момент реализовано немного действий, однако уже сейчас для каждого действия есть определенный набор альтернатив. Таким образом, получилась проблема принятия решений «модульным человеком». Для решения этой проблемы были введены факторы

(временные, энергетические), цели, эмоции, интересы, результаты действий и прочее.

Стоит отметить, что помимо моделирования отдельного человека, модель предполагает моделирование групп людей, взаимодействующих друг с другом. Это введено для более корректного описания человека, ведь человек социален. Также для социализации человека в модели введены фирмы, на которых человек может работать.

К сегодняшнему дню выпущена демонстрационная версия «Модели поведения человека». Подходы, использованные в модели, тяжело классифицировать. Можно сказать, что некоторые методы являются концептуально новыми. Это отрицательно сказывается на их понимании и возможности внедрения, так как их формальное описание, а также публикации о достоинствах и недостатках их использования, не представлены.

В результате можно сделать вывод, что разработки активно ведутся во многих направлениях, применяются различные подходы для реализации необходимого функционала. Однако многие разработки применимы исключительно в их предметной области. Также большинство программных реализаций опирается лишь на текущее состояние индивида, что является минусом, по сравнению с системами, учитывающими его прошлое.

Основные концепции, применимые для построения модели поведения человека и группы лиц

Анализируя вышеуказанные исследования и разработки, можно утверждать, что моделирование поведения человека применимо к различным областям знаний. Однако данные исследования зачастую направлены на довольно узкий круг задач и применение изложенных методов практически невозможно вне этого круга. Также следует заметить, что рассмотренные модели поведения человека крайне слабо учитывают его прошлое, а также делают слабый акцент на текущем психо-, социо- и физиологическом состоянии.

Это заключение позволяет говорить о необходимости создания нового проекта для моделирования поведения, в котором будут отсутствовать основные недостатки приведенных выше систем. Все еще остаются нерешенными вопросы моделирования адаптивной к разным задачам системы, способной прогнозировать деятельность человека.

Для реализации этой идеи необходимо создание модели, способной адекватно учитывать любые факторы, влияющие на поведение человека. В частности, необходимо учитывать события, произошедшие с человеком до моделируемого события, текущие психологическое и физиологическое состояние, социальное положение и социальные связи, и прочее. Рассмотрим идею более детально.

Поведение – способность человека и животных изменять свои действия, реагировать на

воздействие внутренних и внешних факторов [11]. Данное множество факторов, в комплексе, и определяют вероятность конкретных действий:

$$\text{Действие} = F(\text{фактор}1, \text{фактор}2, \dots, \text{фактор}N)$$

Под фактором в данном случае может пониматься характеристика человека, событие, социальная связь и прочее. Множество действий, в свою очередь, определяют общее поведение человека:

Соответственно, чем больше факторов и вариантов действий можно представить в модели поведения, тем точнее будет прогноз деятельности человека (при условии корректного учета влияния каждого фактора на шанс совершения поступка).

При успешном моделировании деятельности одного человека получаем возможность моделирования поведения групп лиц как результат деятельности множества людей. При этом не следует забывать о важности моделирования связей между людьми.

Таким образом, для построения модели необходимо:

1. Установить способ представления человека как множества персонализированных характеристик (факторов, влияющих на его деятельность\поведение).
2. Выделить факторы, влияющие на деятельность\поведение человека (в частности, с точки зрения правонарушений).
3. Определить зависимости (степень влияния) отдельных факторов на каждый вариант действий.
4. Определить оптимальный метод выбора действий на основании факторов.
5. Учесть в модели взаимное влияние индивидов друг на друга.

В этом случае человека необходимо представить, как обособленную единицу с индивидуальным набором характеристик и возможностью связи с другими единицами. Оптимальным подходом для этого является использование агентного моделирования с представлением человека в качестве агента.

Агентное моделирование – это раздел имитационного моделирования, где для создания поведения целой системы разрабатываются элементы системы – взаимодействующие, но независимые агенты. Поведение системы в этом случае возникает как результат деятельности агентов. В качестве агентов допускается использование практически любых объектов. Однако они должны соответствовать необходимым нормам «интеллектуальности». Например, на рисунке 1, изображен пример общей внутренней организации агента.

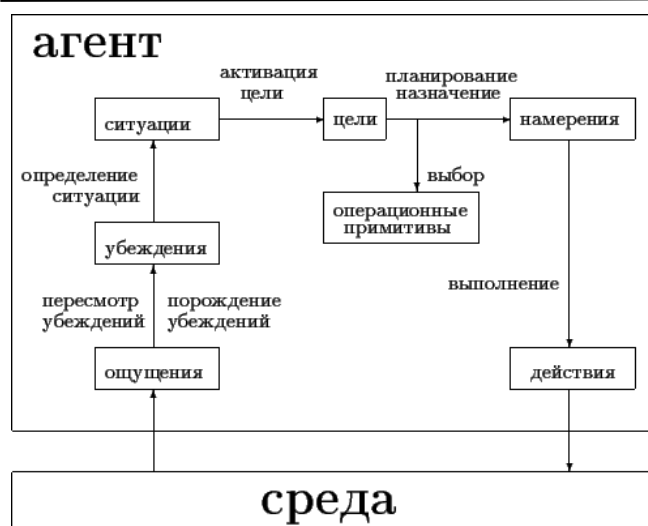


Рисунок 1 – Схема реализации эмоционально-мотивированного агента в мультиагентной системе моделирования поведения

В отличие от системной динамики и дискретно-событийных моделей агентные модели децентрализованы. Здесь не определяется поведение системы в целом, поведение агентов определяется на индивидуальном уровне, а динамика системы возникает как результат деятельности многих агентов. Агентное моделирование является подходом более универсальным и мощным, т. к. оно позволяет учесть любые сложные структуры и поведения. Другое важное преимущество агентного моделирования в том, что разработка модели возможна и при отсутствии знания о глобальных зависимостях: нужно определять индивидуальную логику поведения участников процесса для того, чтобы построить агентную модель и использовать ее для имитации глобального поведения всей системы. Агентную модель проще поддерживать: уточнения обычно делаются на локальном уровне и не требуют глобальных изменений.

Мультиагентные системы состоят из множества искусственных агентов, которые работают совместно. Агент – это эволюция понятия «объект». Существует много определений агента. Общим в них является то, что агент – некоторая сущность, которая обладает активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, может взаимодействовать с окружением и другими агентами и может изменяться. Можно сказать, что агент – это объект, обладающий определенными свойствами. Приведем основные свойства, присущие агенту:

- автономность: агенты функционируют без прямого вмешательства в их действия, они могут самостоятельно контролировать свое состояние и реагировать на происходящие изменения;
- методы общения: агенты взаимодействуют друг с другом с помощью некоторого языка;
- реактивность: агенты способны воспринимать окружающую среду;

– активность: агенты обладают целенаправленным поведением и способны сами проявлять инициативу;

– интеллектуальное поведение: агент способен к обучению, способен находить оптимальные способы поведения;

– индивидуальная картина мира: каждый агент по-своему воспринимает окружающую среду;

– мобильность: способность к передаче кода агента.

Для того чтобы агент мог вести себя определенным образом, он должен иметь специальные «устройства»: эффекторы – органы, воздействующие на среду, рецепторы – органы, получающие информацию от воздействия среды, и процессор для обработки информации.

Для построения агентной модели необходимо определить множество агентов и основы их поведения, определить взаимоотношения между агентами и теоретические основы этих отношений, выбрать платформу для агентного моделирования. Определение агентов с точным заданием их поведения и взаимодействия с другими агентами образует базу для разработки адекватной агентной модели.

Предположительно количество характеристик человека весьма велико и прямо пропорционально достоверности результата моделирования. Это означает, что для хороших результатов может понадобиться большое количество данных о каждом человеке. Чтобы корректно учитывать различные и разноразмерные характеристики, а также получить оптимальную деятельность агента, возможно использование искусственной нейронной сети.

Искусственные нейронные сети (ИНС) – математические модели, а также их программные или аппаратные реализации, построенные по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге при мышлении, и при попытке смоделировать эти процессы. Первой такой моделью мозга был перцептрон. Впоследствии эти модели стали использовать в практических целях, как правило, в задачах прогнозирования.

ИНС представляют собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие локально простые процессоры вместе способны выполнять довольно сложные задачи.

С математической точки зрения, обучение нейронных сетей – это многопараметрическая задача нелинейной оптимизации. С точки зрения кибернетики, нейронная сеть используется в задачах

адаптивного управления и как средство реализации алгоритмов для робототехники, а с точки зрения искусственного интеллекта, ИНС является основным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью компьютерных алгоритмов.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке.

Решение задачи с применением ИНС включает несколько этапов:

- Сбор данных для обучения;
- Подготовка и нормализация данных;
- Выбор топологии сети;
- Экспериментальный подбор характеристик сети;
- Экспериментальный подбор параметров обучения;
- Собственно обучение;
- Проверка адекватности обучения;
- Корректировка параметров, окончательное обучение.

На вход сети, в случае моделирования поведения, будут подаваться характеристики человека, приведенные к общей размерности. Необходимо обеспечить учет характеристик вместе с их влиянием на индивида (весами каждой характеристики). Выходами сети будет множество допустимых действий человека.

Комбинирование агентного подхода и нейронных сетей для получения состояния агента, при условии наличия качественных данных о характеристиках человека и корректной модели социальных связей, может дать хороший результат. Особенно интересным здесь является то, что чем больше характеристик мы внесем, тем точнее будет прогноз поведения, в то время, когда сами характеристики могут относиться к любой сфере жизнедеятельности человека. Это значит, что модель будет применима для моделирования поведения групп лиц практически в любой ситуации.

Заключение

Исследуемое в статье направление имеет огромный потенциал и, соответственно, хорошие перспективы. Любые разработки, направленные на исследование и прогнозирование человеческих реакций и действий будет востребовано во многих областях знаний, а также многими фирмами и корпорациями.

Сегодня программные реализации моделирования поведения человека уже довольно обширны. Но, все же, их недостаточно для реализации многих задач и учета полного спектра характеристик индивидуума.

Анализируя предметную область, можно говорить о возможности проведения новой разработки прогнозирования поведения групп лиц на основе агентного моделирования и нейронных сетей. В этом случае часть нерешенных вопросов общей проблемы моделирования поведения человека сможет быть решена.

Литература:

1. Карпов В.Э. Коллективное поведение роботов. Желаемое и действительное // Современная мехатроника. Сб. научн.трудов Всероссийской научной школы (г.Орехово-Зуево, 22-23 сентября 2011), Орехово-Зуево, 2011, с.35-51, -132с.
2. Петрушан М.В., Самарин А.И. Эволюционная модель поведения агентов в среде, побуждающей к коллективным действиям // Научная сессия МИФИ-2006. Нейроинформатика - 2006. Часть 1 Теория нейронных сетей. Нейробиология. Нейронные сети в задачах обработки изображений. Нейробиология. Нейронные сети в задачах обработки изображений. Модели адаптивного поведения, стр. 196-200
3. VT MAK – Project Humans – DI-Guy SDK [Электронный ресурс]. – Режим доступа: <http://www.mak.com/products/humans>.
4. Макаров О. Игра в людей: Цифровой аналог человечества // Популярная механика. №3 (89), 2010 – с. 78 – 80.
5. The Naval Postgraduate School's MOVES Institute [Электронный ресурс]. – Режим доступа: <https://www.movesinstitute.org/>.
6. Chris Darken: Human Behavior Simulation [Электронный ресурс]. – Режим доступа: <https://www.movesinstitute.org/research/human-behavior-simulation/>.
7. Marco Strano and Roberta Bruzzone, Modeling of Human Behavior in Violent Crimes [Электронный ресурс]. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?sessionid=A4A285B0DEB8C9E33391781795309D1E?doi=10.1.1.125.4662&rep=rep1&type=pdf>.
8. Инструмент имитационного моделирования AnyLogic: обзор [Электронный ресурс]. – Режим доступа: <http://www.anylogic.ru/overview>.
9. Бекларян А. Л., Акопов А. С. Моделирование поведения толпы на основе интеллектуальной динамики взаимодействующих агентов / «Бизнес-информатика» / № 1 (31) / 2015 / с 69-77
10. Текущее состояние модели поведения человека [Электронный ресурс]. – Режим доступа: <http://www.humanmodel.ru/model/47-current-model?showall=&start=1>.

11. Биологический энциклопедический словарь. Определение «Поведение» [Электронный

ресурс]. – Режим доступа:
http://dic.academic.ru/dic.nsf/dic_biology/4378.

Землянская С.Ю., Слозеницын А.В., Андриевская Н.К. Структура модели для прогнозирования поведения групп лиц. Рассмотрено общее состояние разработок в области моделирования поведения человека и поведения сообществ. Выявлены достоинства и недостатки существующих моделей и готовых программных комплексов, с точки зрения прогнозирования поведения групп лиц. Определены основные критерии, используемые для построения новой модели поведения человека и прогнозирования поведения групп лиц, лишенной недостатков проанализированных продуктов. Выбраны методы и подходы для создания новой модели, описана ее структура.

Ключевые слова: моделирование поведения, прогнозирование деятельности, искусственный интеллект, агентное моделирование, искусственная нейронная сеть.

Zemlyansky S.Y., Slozhenitsyn A.V., Andrievskaya N.K. Model structure for the people group behavior forecasting. General state of human behavior and crowd behavior modeling is reviewed. Advantages and disadvantages of existing models and softwares on their bases are discovered considering behavior forecasting. The basic criteria of human behavior new model building are revealed without shortcomings of previous models. The methods and approaches for new model constructing are chosen.

Keywords: behavior modeling, activities forecasting, artificial intelligence, agent-based modeling, artificial neural network.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.421.4, 004.421.6

Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС «Android»

И.А. Коломойцева, Т.В. Завадская
Донецкий национальный технический университет
bolatiger@gmail.com

Коломойцева И.А., Завадская Т.В. Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС Android». В данной работе рассматривается решение задачи метапоиска с помощью ГРИД-системы для предметной области «Программирование для ОС Android». Приведена общая схема ГРИД-системы для решения этой задачи. Предложена структура тезауруса для представления естественно-языковой информации, используемой для синтеза запросов метапоисковой системы, и формирования результата её работы. Описана реализация отправки запросов и получения результатов для поисковых машин Google, Yandex, Yahoo, Bing.

Ключевые слова: метапоиск, ГРИД-система, тезаурус, Android, объекты, семантические отношения, Google, Yandex, Yahoo, Bing.

Введение

Наше время можно назвать эпохой мобильных устройств. Они вошли в нашу жизнь и стали её неотъемлемой частью. Большинство мобильных устройств работают под управлением операционной системы Android. По данным сайта <http://4pda.ru> их доля на рынке превысила 86%.

Растёт также количество Android-разработчиков. Основную информацию о программировании под Android они берут из Интернета. Учитывая разнообразие мобильных устройств (смартфоны, планшеты, часы, очки и т.п.), их широкие аппаратные возможности (наличие GPS, акселерометров, NFC и т.п.), частоту выхода версий ОС Android (в среднем раз в два-три месяца), разработчикам приложений приходится очень часто обращаться с запросами к всемирной паутине. С другой стороны, существует огромное количество Интернет-ресурсов, содержащих сведения о программировании для ОС Android. Информация на них часто повторяется и, чтобы найти нужные сведения иногда приходится тратить много времени, просматривая ресурсы, перефразируя запросы, обращаясь к разным поисковым машинам.

Облегчить Android-разработчикам поиск сведений об особенностях программирования под Android может метапоисковая система.

Метапоисковая система (метапоисковая машина) — это поисковая система, которая в отличие от классических поисковых машин не имеет собственной базы данных и собственного

поискового индекса, а формирует поисковую выдачу за счет смешивания и переранжирования результатов поиска других поисковых систем [2].

Среди разработанных метапоисковых систем следует отметить Web Scout (ищет новости, конференции, аукционы), 1 SECOND, search.da.ru, exactus [1].

Задача метапоиска ресурсозатратная, обладает естественным распараллеливанием. Кроме этого, при реализации автоматических запросов к стандартным поисковым машинам возникают проблемы, связанные с ограничением, установленным этими машинами. Они ограничивают количество автоматических запросов, которые можно выполнить в единицу времени с одного IP-адреса.

Снять это ограничение и увеличить скорость работы метапоисковой системы можно помощью ГРИД-системы.

Направление развития ГРИД-технологии происходит по трём направлениям [2]:

- 1) вычислительные ГРИД;
- 2) семантические ГРИД;
- 3) ГРИД для интенсивной обработки данных.

ГРИД первого направления предназначены для решения вычислительных задач, например с использованием численных методов (решение уравнение математической физики) [2].

Семантические ГРИД предназначены для оперирования данными из различных баз данных.

ГРИД третьего направления предназначены для обработки огромных объёмов данных несложными программами, которые можно

реализовать на персональном компьютере. Сложность этого направления – доставка данных для обработки и пересылка результатов [2].

ГРИД-системы используются для решения задач прогноза погоды, обработка космических, астрофизических, геофизических данных и для БАК, индексирование новостных БД, в физике высоких энергий, биомедицинских науках, вычислительной химии, ядерного синтеза, финансово-экономических исследований [2, 3, 4, 5].

Задача метапоиска информации о программировании для ОС Android требует обработки большого объема информации. Поэтому её решение с помощью ГРИД-системы является актуальной задачей.

Целью данной статьи является описание структурной схемы ГРИД-системы, решающей задачу метапоиска информации о программировании для ОС Android, структуры текстовой информации, которой она оперирует, а также описание использования поисковых машин Google, Yandex, Yahoo и Bing в рассматриваемой

системе.

Общая постановка задачи

ГРИД-система для решения задачи метапоиска делится на следующие подсистемы (рисунок 1):

- подсистема интерфейса пользователя, которая обеспечивает ввод запроса от пользователя и вывод результатов поиска;
- подсистема генерации запросов, которая по введенному пользователем генерирует запросы, семантически идентичные исходному;
- подсистема работы брокера (планировщика), которая распределяет запросы между узлами и принимает результат работы;
- подсистема работы узлов Грид-системы, выполняющих поиск;
- подсистема отбора релевантных результатов, которая получает их от брокера и после обработки передает подсистеме интерфейса пользователя.

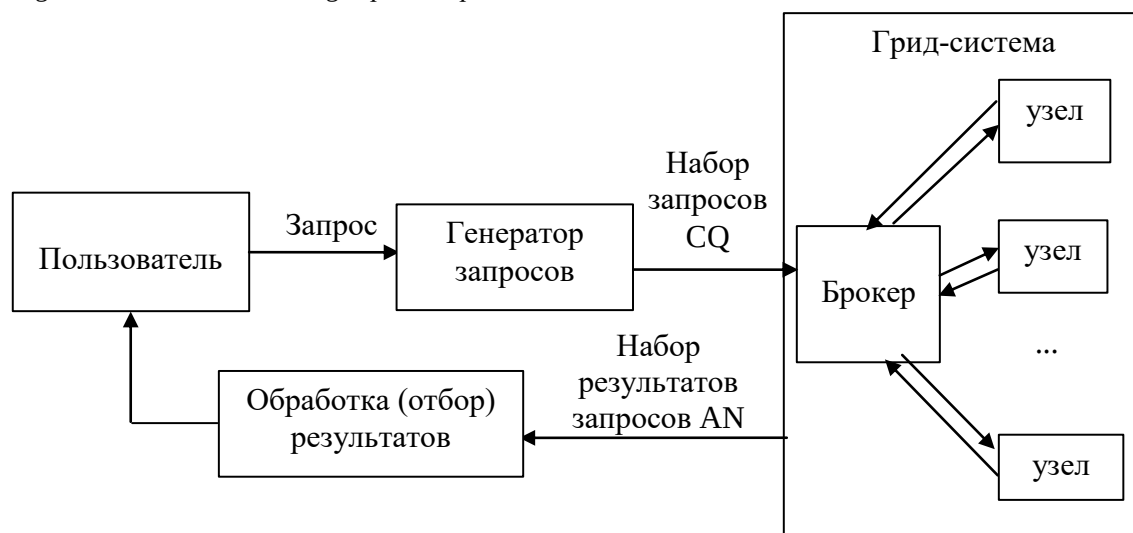


Рисунок 1 – Общая схема ГРИД-системы, решающей задачу метапоиска

Базой для генерации запросов для подсистемы генерации запросов является тезаурус. Тезаурус обычно привязан к некоторой предметной области. Например, в работе [6] предложена объектная модель семантического анализа естественно-языкового текста, используемая для тезауруса запросов в таких предметных областях, как “заболевания” [6] и “лекарства” [7]. Описание тезауруса для предметной области «Программирования для ОС Android» приведено в этой статье дальше.

Узлы ГРИД-системы при решении задачи метапоиска могут выполнять две задачи:

- запускать стандартную поисковую

машину с конкретными запросами и получать результат;

- в БД текстов, которая хранится на узлах, осуществлять поиск по конкретному запросу и выдавать список документов, ему удовлетворяющих.

В [8, 9] предлагается структура подсистемы, которая на основе объектной модели из [6], выполняет анализ текстовой БД и отбирает нужные документы. Отобранные документы вместе с результатами работы стандартных поисковых машин, отправляются в подсистему отбора релевантных результатов.

Отбор результатов в подсистеме отбора релевантных результатов может осуществляться

на основе двух подходов:

- объектного;
- функционального.

В работах [6-9] описан метод отбора релевантных результатов на основе анализа выдач с помощью тезауруса, построенного посредством объектной модели.

Альтернативой является функциональный подход, представленный в работе [9]. В соответствии с этим подходом выдачи результатов запросов можно представить в виде функций, аргументы которых должны быть сопоставлены с образцами. Однако этот подход имеет более высокую вычислительную сложность.

Алгоритм работы брокера приведен в [10].

Объекты и семантические отношения в тезаурусе «Программирование для ОС Android»

Чтобы использовать естественный язык в качестве основы для построения языка представления знаний, в нем предлагается выделить несколько классов-элементов. Эти классы можно разделить на две категории: семантически значимые объекты предложения и семантические отношения. Объекты еще называют именами [11] и именованными сущностями [12]. Примеры объектов, представленных в ЕЯ-текстах с описанием элементов программирования для ОС Android,

приведены в таблице 1.

Часть из приведенных в таблице 1 объектов имеют сложную иерархическую структуру. К таким объектам можно отнести ВИДЖЕТЫ. Из таблицы 1, например, видно, что к ВИДЖЕТАМ относятся ИНДИКАТОРЫ, а ИНДИКАТОРЫ, в свою очередь, делятся на ProgressBar, RatingBar, SeekBar. Имеют свои разновидности СПИСКИ, ТЕКСТОВЫЕ_ПОЛЯ, КНОПКИ и другие ВИДЖЕТЫ.

Объекты связываются между собой с помощью семантических отношений. Выдвинута гипотеза, согласно которой множество отношений, в отличие от множеств объектов (имен), конечно [11]. В [11] выделено около 200 не сводимых к друг другу отношений. В [13] 200 отношений из [11] сведены к семнадцати.

Классы-объекты можно представить в виде древовидной структуры. Особенностью данного дерева является то, что в узлах дерева находятся названия классов, а листьями являются понятия данного класса, что позволяет достаточно четко их определять. Все листья, которые определены в данном классе, являются синонимами.

В естественно-языковых текстах из предметной области «Программирование для ОС Android» можно выделить следующие семантические связи: генеративную, результативную, инструментальную, каузальную, комитативную [13].

Таблица 1. Примеры объектов, представленных в ЕЯ-текстах с информацией о программировании для ОС Android

№ п/п	Название объекта	Примеры объектов
1	КОМПОНЕНТ	Деятельность (Activity), служба (Service), приёмник широковещательных намерений (Broadcast Receiver), контент-провайдер (Content Provider)
2	СТАТУС_ПРОЦЕССА	Активный процесс, фоновый процесс, пустой процесс
3	РАЗРЕШЕНИЕ	Permission-tree, permission-group
4	РЕСУРС	Изображение, звук
5	РАЗМЕТКА	Фрейм (FrameLayout), линейная (LinearLayout), табличная (TableLayout), относительная (RelativeLayout)
6	ВИДЖЕТ	Текстовые поля, списки, кнопки, индикаторы
7	ИНДИКАТОР	ProgressBar, RatingBar, SeekBar
8	УВЕДОМЛЕНИЕ	Вплывающее уведомление, уведомление в строке состояния
9	ДИАЛОГОВОЕ_ОКНО	AlertDialog, DatePickerDialog, TimePickerDialog, ProgressDialog
10	МЕНЮ	Меню опций, контекстное меню, подменю
11	ДАТЧИК	SensorManager
12	ПЕРЕДАЧА ДАННЫХ	Bluetooth, Wi-Fi, NFC
13	ТЕЛЕФОНИЯ	Звонок, SMS
14	ДАННЫЕ	SQLite, геолокационные данные

Генеративная связь имеет место, когда один компонент обозначает лицо или предмет, принадлежащий некоторой совокупности, категории, обозначаемой вторым компонентом.

Результативная присутствует в тех предложениях, где один компонент выражает следствие действия второго.

Инструментальная означает, что один компонент обозначает орудие действия, обозначаемого другим компонентом.

Каузальная имеет место, когда один

компонент обозначает причину появления другого компонента спустя какое-то время.

Комитативная встречается в тех предложениях, когда один компонент обозначает сопровождающее другой компонент действие, сопутствующий предмет, сопровождающее лицо.

Примеры объектов естественно-языковых текстов из предметной области «Программирование для ОС Android», связываемых семантическими отношениями, представлены в таблице 2.

Таблица 2. Семантические отношения и связываемые ими объекты

Семантическая связь	Связываемые объекты
Результативная	СОБЫТИЕ → ДЕЙСТВИЕ
Инструментальная	РАЗРЕШЕНИЕ → ДЕЯТЕЛЬНОСТЬ
Каузальная	УВЕДОМЛЕНИЕ → ДЕЯТЕЛЬНОСТЬ
Комитативная	МЕНЮ → ПОДМЕНЮ

Описание тезауруса предметной области «Программирование для ОС Android»

Знания о предметной области «Программирование для ОС Android» необходимо представить в виде, пригодной для автоматической обработки. Онтологии являются именно такой формой представления знаний.

Онтология – соглашение об общем использовании понятий, которое содержит средства представления предметных знаний и договоренности о методах соображений. Она может рассматриваться как определенное описание взгляда на мир в конкретной сфере интересов, который состоит из набора терминов и правил использования этих терминов, которые ограничивают их значение в рамках конкретной предметной области [11].

Согласно «Современному словарю иностранных слов» тезаурус определяется как полный систематизированный набор данных о какой-либо области знаний, позволяющий человеку или вычислительной машине в ней ориентироваться. Таким образом, тезаурус можно рассматривать как частный случай онтологии.

Существует множество стандартов на формат представления тезаурусов. Основными документами, регламентирующими формат представления тезауруса, являются стандарты ISO 2788-1986 для описания одноязычных тезаурусов и ISO 5964-1985. При разработке тезауруса предметной области «Программирование для ОС Android» будем опираться на первый из перечисленных стандартов.

Стандарт ISO 2788-1986 определяет тезаурус как набор терминов, связанных между собой соответствующими отношениями.

Каждый термин характеризуется комментарием и ссылкой на понятие верхнего уровня.

Основные понятия, представленные в тезаурусе «Программирование для ОС Android»: компонент, статус_процесса, разрешение, ресурс, разметка, виджет, уведомление, диалоговое_окно, меню, датчик, передача_данных, телефония, данные.

Связи, которые представлены в тезаурусе «Программирование для ОС Android»:

– USE – связь с очень близким по смыслу синонимом (термины А и В, представляющие некоторое понятие, обозначают практически одно и то же; например, у одного виджета может быть несколько названий, как минимум английское и русское);

– UF – связь, обратная USE;

– BT – связь термина с понятием высокого уровня (например, термин «текстовое поле» есть уточнение понятия «виджет»);

– NT – связь, обратная BT;

– Res – результативная;

– Ins – инструментальная;

– Caus – каузальная;

– Com – комитативная.

Программно тезаурус представлен таблицами termin, comment и links.

Таблица termin содержит поля, в которых присутствует информация о названии понятия, ссылка на комментарий, ссылка на информацию о связях, идентификатор уровня.

Таблица comment содержит словесное представление комментария.

Таблица links содержит информацию о типе связи, идентификаторы связываемых понятий.

Разработанный тезаурус можно

использовать для генерации запросов и формирования окончательного результата для вывода пользователю, в метапоисковой системе для предметной области «Программирование для ОС Android».

Использование существующих поисковых систем при решении задачи метапоиска

При решении задачи метапоиска для собственно поиска информации в сети Интернет используются стандартные поисковые машины. В метапоисковой системе «Программирование для ОС Android» с качестве таких поисковых машин используются Google, Yandex, Bing и Yahoo. Эти поисковые машины популярны в мире и предоставляют программные средства для организации автоматического выполнения запросов и получения их результатов.

Метапоисковой системе «Программирование для ОС Android» на вход

подаётся запрос от пользователя. Используя тезаурус, описанный выше, генерируется N похожих по смыслу запросов. Количество запросов N зависит от числа слов в запросе, количества синонимов в тезаурусе для каждого из слов и типа семантической связи между словами запроса. Каждый из этих N запросов запускается на всех поисковых машинах: Google, Yandex, Bing и Yahoo. Эти 4*N запросов брокер ГРИД-системы равномерно распределяет между доступными в текущий момент узлами системы. Подробно алгоритм распределения запросов между узлами ГРИД-системы описан в [10].

Опишем как организован программной генерации запроса и получения результата в каждой из поисковых машин.

Google использует методы GET или POST при отправлении запроса, результат формируется с помощью технологии REST. Функция организации поиска для поисковой машины Google приведена на рисунке 2.

```
public void Search()
{
    int count = 0;
    while(count < 32)
    {
        HttpWebRequest Request = BuildRequest(Query, count);
        HttpWebResponse Response = (HttpWebResponse)Request.GetResponse();
        StreamReader sr = new StreamReader(Response.GetResponseStream());
        JObject responseObj = JObject.Parse(sr.ReadToEnd());
        JToken responseData = (JToken)responseObj["responseData"];
        JArray resultArray = (JArray)responseData["results"];
        for (int i = 0; i < resultArray.Count; i++)
        {
            string cnt = (string)responseData["cursor"]["estimatedResultCount"];
            Uri url = new Uri((string)resultArray[i]["unescapedUrl"]);
            Results.Add(new
            SearchResult(BoldText((string)resultArray[i]["title"]),
            url.ToString(), BoldText((string)resultArray[i]["content"]), (string)resultArray[i]["visible
            Url"], "Google", Convert.ToInt32(cnt), Rating));
        }
        count += 8;
    }
    Results.RemoveRange(30, 2);
    isSearching = false;
}
```

Рисунок 2 – Функция поиска Google

```
<?xml version="1.0" encoding="windows-1251"?>
<request>
  <query>yandex</query>
  <groupings>
    <groupby attr="d" mode="deep" groups-on-page="10" docs-in-group="1"
  />
  </groupings>
</request>
```

Рисунок 3 – Пример простого запроса

Yandex использует технологию Яндекс.XML. Пример простого запроса приведен на рисунке 3.

Yahoo использует технологии REST и SOAP для составления запросов и получения результатов. Разработчиками поисковой машины

предоставлена библиотека, позволяющая реализовать поиск в Интернете, изображений, видео и другой полезной информации практически на любом языке программирования.

Функция организации поиска для поисковой машины Yahoo приведена на рисунке 4.

```
public void Search(string Query)
{
    YahooSearch = new YahooSearchService();
    Yahoo.API.WebSearchResponse.ResultSet resultSet =
    YahooSearch.WebSearch(AppId, Query, "all", 30, 1, "all", false, false, "ru");
    foreach (Yahoo.API.WebSearchResponse.ResultType result in
    resultSet.Result)
    {
        Results.Add(new
        SearchResult(result.Title, result.Url, result.Summary));
    }
}
```

Рисунок 4 – Функция поиска для поисковой машины Yahoo

```
private HttpWebRequest BuildRequest(string Query)
{
    string requestString = "http://api.search.live.net/xml.aspx?"
    // Common request fields (required)
    + "AppId=" + AppId
    + "&Query=" + Query
    + "&Sources=Web"
    // Common request fields (optional)
    + "&Version=2.0"
    + "&Market=ru-ru"
    + "&Adult=Moderate"
    + "&Options=EnableHighlighting"
    + "&Web.Count=30"
    + "&Web.Offset=0"
    +
    "&Web.Options=DisableHostCollapsing+DisableQueryAlterations";
    HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(
    requestString);
    return request;
}
```

Рисунок 5 – Функция построения запроса для поисковой машины Bing

```
private List<SearchResult> GetResults(XmlNode root, XmlNamespaceManager nsmgr)
{
    List<SearchResult> SResults = new List<SearchResult>();
    SearchResult SResult = new SearchResult();
    // Add the Web SourceType namespace to the namespace manager.
    nsmgr.AddNamespace(
    "web",
    "http://schemas.microsoft.com/LiveSearch/2008/04/XML/web");
    XmlNode web = root.SelectSingleNode("./web:Web", nsmgr);
    XmlNodeList results = web.SelectNodes(
    "./web:Results/web:WebResult",
    nsmgr);
    foreach (XmlNode result in results)
    {
        SResult.Title = result.SelectSingleNode("./web:Title", nsmgr).InnerText;
        SResult.URL = result.SelectSingleNode("./web:Url", nsmgr).InnerText;
        SResult.Body = result.SelectSingleNode("./web:Description", nsmgr).InnerText;
        SResults.Add(new SearchResult(SResult.Title, SResult.URL, SResult.Body));
    }
    return SResults;
}
```

Рисунок 6 – Получение результатов поиска для поисковой машины Bing

Bing использует технологии JSON, XML и SOAP. Функция построения запроса для

поисковой машины Bing приведена на рисунке 5, а функция получения результата – на рисунке 6.

Результаты работы поисковых машин, полученные брокером от узлов ГРИД-системы, передаются подсистеме формирования ответа, которая формирует иговый html-документ для предоставления пользователю.

Выводы

В данной работе рассмотрена организация ГРИД-системы, решающей задачу метапоиска для предметной области «Программирование для ОС Android». В частности выделены основные её структурные элементы.

Разработан тезаурус для предметной области «Программирование для ОС Android». Этот тезаурус будет использован для синтеза запросов метапоисковой системы и для формирования результата её работы, который будет предоставлен пользователю.

Литература

1. Пасько В.П. Энциклопедия ПК. Аппаратура. Программы. Интернет. – Киев, издательская группа БНВ; СПб: Питер, 2004. – 800 с.
2. Котляр В.В. Применение Грид-технологий для задач интенсивной обработки данных – Режим доступа: <http://litcey.ru/geografiya/5431/index.html>
3. Castellano M. Biomedical Text Mining Using a Grid Computing Approach / M. Castellano, G. Mastronardi, Decataldo G., Pisciotto L., Tarricone G., Cariello L., Bevilacqua V. – Режим доступа <http://cdn.intechopen.com/pdfs-wm/12928.pdf>.
4. Hughes B. Grid-based Indexing of a Newswire Corpus / B. Hughes, S. Venugopal, R. Buyya. – Режим доступа: <http://www.cloudbus.org/papers/nlp-newswire-grid.pdf>.
5. Li Q. The Future-Oriented Grid-Smart Grid / Q. Li, M. Zhou // Journal of computers, Vol 6. – №1. – 2011. – p.98-105.
6. Коломойцева И.А. Объектная модель семантического анализа естественно-языкового медицинского текста / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2007). Выпуск 8 (120). – Донецк: ДонНТУ, 2007. –

С. 141-150.

7. Коломойцева И.А. Компьютерный тезаурус для предметной области «ЛЕКАРСТВА» / И.А. Коломойцева // Моделирование и компьютерная графика: Материалы 4-й международной научно-технической конференции, г. Донецк, 5-8 октября 2011 г. — Донецк, ДонНТУ, Министерство науки и образования, молодёжи и спорта, 2011. – с. 161-165.
8. Коломойцева И.А. Объектная модель естественно-языкового медицинского текста на примере системы «ФармАналитик» / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2009). Выпуск 10 (153). – Донецк: ДонНТУ, 2009. – С. 303-306.
9. Коломойцева И.А. Функциональная модель медицинского естественно-языкового текста / И.А. Коломойцева // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2008). Выпуск 9 (132). – Донецк: ДонНТУ, 2008. – С. 237-241.
10. Коломойцева И.А., Дацун Н.Н. Алгоритм работы брокера ГРИД-системы для решения задачи метапоиска // Научные труды Донецкого национального технического университета. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ-2014). – Донецк: ГВУЗ «ДонНТУ», 2014. – Вып. 1 (19) – С. 92-96.
11. Поспелов Д. А. Логико-лингвистические модели в системах управления. М.: Энергоиздат, 1981. 232 с.
12. Хорошевский В.Ф. Оценка систем извлечения информации из текстов на естественном языке: кто виноват, что делать // Труды Десятой национальной конференции по искусственному интеллекту с международным участием (КИИ-2006). М.: Физматлит, 2006. Т. 2. С. 464-478.
13. Осипов Г.С. Приобретение знаний интеллектуальными системами: Основы теории и технологии. М.: Наука. Физматлит, 1997. 112 с.

Коломойцева И.А., Завадская Т.В. Особенности поиска информации в метапоисковой ГРИД-системе «Программирование для ОС Android». В данной работе рассматривается решение задачи метапоиска с помощью ГРИД-системы для предметной области «Программирование для ОС Android». Приведена общая схема ГРИД-системы для решения этой задачи. Эта схема включает подсистемы пользовательского интерфейса и генерации запросов, брокер, узлы и подсистему генерации результатов. Подсистема пользовательского интерфейса предназначена для взаимодействия с пользователем. Подсистема генерации запросов создаёт множество запросов на основе пользовательского запроса, используя описанный в статье тезаурус. Брокер распределяет запросы между узлами ГРИД-системы и получает от них результаты работы. Подсистема генерации результатов отбирает среди полученных данных работы ГРИД-системы

релевантные пользовательскому запросу ответы. Предложена структура тезауруса для представления естественно-языковой информации, используемой для синтеза запросов метапоисковой системы, и формирования результата её работы. Описана реализация отправки запросов и получения результатов для поисковых машин Google, Yandex, Yahoo, Bing.

Ключевые слова: метапоиск, ГРИД-система, тезаурус, Android, объекты, семантические отношения, Google, Yandex, Yahoo, Bing.

.Kolomoitseva I.A., Zavadskaja T.V. Information Search Features in metasearch GRID-system "Programming for Android". This article describes the solution metasearch tasks using GRID-system for domain "Programming for Android". The General scheme of the GRID to solve this problem was developed. This scheme includes the subsystem of user interface, the subsystem of generating of requests, the subsystem of the broker, the subsystem of nodes, the subsystem of selection of relevant results. The subsystem of user interface is designed for user interaction with the grid system. The subsystem of request generation creates a lot of queries by the user's request. The subsystem of the broker distributes requests between the nodes and takes the result of the work. The subsystem of nodes executes a search using the standard search engines. The subsystem selection of relevant results selects relevant results from a variety of responses that came from the broker. The structure of a thesaurus to represent natural language information was developed. This thesaurus used for the synthesis queries of metasearch system, and the formation of the result of work this system. This article describes the implementation of sending requests and receiving the results of the Google search engine, Yandex, Yahoo, Bing.

Keyword: metasearch, GRID-system, thesaurus, Android, objects, semantic relations Google, Yandex, Yahoo, Bing.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.42

Информационная система учета кадров организации

А.И. Костюк, Е.О. Лебедева
Южный федеральный университет
aikostyuk@sfedu.ru

Костюк А.И., Лебедева Е.О. Информационная система учета кадров организации. Предложена информационная система учета кадров организации, автоматизирующая работу работников отдела кадров, разработанная на основе контекстной диаграммы, диаграммы декомпозиции, диаграмм логической и физической моделей базы данных. Разработанная информационная система позволяет повысить эффективность автоматизации документооборота за счет существенного снижения времени, необходимого для обработки информации и оформления кадровых документов.

Ключевые слова: автоматизированное рабочее место, информационная система, база данных, отдел кадров, программное обеспечение.

Введение

Одним из факторов, определяющих уровень развития современного общества и его интеллектуальные возможности, является оснащённость его средствами вычислительной техники – основы автоматизации умственной деятельности человека. Сфера использования ЭВМ в настоящее время настолько широка, что нет такой области, где применение ЭВМ было бы нецелесообразным. В настоящее время ЭВМ широко применяются во многих отраслях деятельности человека. Ни одна фирма не может обойтись в своей работе без применения компьютеров, которые с успехом заменяют рутинную работу, выполнявшуюся ранее в ручную, повышая эффективность работы любой фирмы. Особенно важна роль ЭВМ для развития науки, роста промышленного производства и повышения эффективности управления.

Управление персоналом – основным ресурсом каждого предприятия – сегодня становится задачей первостепенной важности. Создание условий для привлечения и удержания лучших сотрудников, своевременная подготовка кадровых ресурсов для удовлетворения потребностей бизнеса – это необходимые составляющие успеха любой компании. Но для того чтобы иметь возможность принимать верные управленческие решения, нужно владеть актуальной, полной и достоверной информацией о персонале, который, к тому же, является наиболее непостоянным, изменчивым ресурсом предприятия. Возможности его использования зависят от многих факторов – экономических, законодательных, демографических. Чтобы

анализировать ситуацию, руководитель должен знать о своих кадрах всё.

Одной из современных тенденций в применении средств автоматизации процессов управления персоналом на крупных предприятиях становится создание единой централизованной информационной системы управления персоналом, которая служит эффективным инструментом для автоматизации процедур HR-отдела, для консолидации и анализа информации о кадровых ресурсах. Единая система позволяет вводить данные однократно и в дальнейшем использовать в разных видах кадровых операций и учета. В централизованной автоматизированной системе поддерживается единое хранилище информации, что дает возможность формировать сводные отчеты и получать интересующую руководителя информацию.

Постановка проблемы

Задачами отдела кадров организации-заказчика являются:

- подбор, расстановка и воспитание кадров;
- изучение деловых и моральных качеств работников по их практической деятельности;
- создание резерва кадров для выдвижения на руководящие и материально ответственные должности;
- организация и проведение всех видов подготовки и повышения квалификации кадров;
- учет кадров;
- обеспечение прав, льгот и гарантий работников организации;

В функции отдела кадров входит:

- разработка кадровой политики и стратегии;
- разработка прогнозов, определение текущей потребности в кадрах и источников ее

удовлетворения на основе изучения рынка труда;

- комплектование организации кадрами служащих и специалистов требуемых профессий, специальностей и квалификации в соответствии с изменяющимися внешними и внутренними условиями деятельности организации;

- формирование и ведение банка данных о количественном и качественном составе кадров;

- подбор и отбор работников совместно с руководителями заинтересованных подразделений и внесение соответствующих предложений об их назначении на указанные должности, оформление приказов о приеме на работу и другой необходимой для этого документации;

- разработка предложений о приеме на работу по конкурсу в соответствии с порядком, установленным законодательством, подготовка и организация работы конкурсной комиссии;

- информирование работников внутри организации об имеющихся вакансиях, использование средств массовой информации для помещения объявлений о найме работников;

- установление прямых связей с учебными заведениями и службами занятости;

- оформление приема, перевода и увольнения работников в соответствии с трудовым законодательством, положениями, инструкциями и приказами руководства организации;

- учет личного состава;

- выдача справок о настоящей и прошлой трудовой деятельности работников;

- прием, заполнение, хранение и выдача трудовых книжек;

- ведение установленной документации по кадрам;

- подготовка материалов для представления персонала к поощрениям

- подготовка материалов по привлечению работников к материальной и дисциплинарной ответственности;

- контроль за правильностью расстановки работников и использования их труда в структурных подразделениях предприятия;

- организация проведения аттестации работников, ее методическое информационное обеспечение, участие в анализе результатов аттестации, осуществление постоянного контроля за ходом выполнения решений аттестационной комиссии;

- подготовка необходимых материалов для рассмотрения на комиссии по установлению выслуги лет;

- подготовка соответствующих документов по пенсионному страхованию и представление их в органы социального обеспечения;

- выдача справок о работе на предприятии, занимаемой должности и размере заработной платы;

- обеспечение социальных гарантий трудящихся в области занятости, соблюдения порядка трудоустройства и переобучения высвобождающихся работников, предоставления им установленных льгот и компенсаций;

- составление графиков отпусков, учет использования работниками отпусков, оформление очередных отпусков в соответствии с утвержденными графиками и дополнительных отпусков;

- оформление и учет командировок;

- табельный учет;

- анализ текучести кадров;

- рассмотрение жалоб и заявлений работников по вопросам приема, перемещения и увольнения, нарушения трудового законодательства;

- принятие мер к выявлению и устранению причин, порождающих жалобы работников.

На начальника отдела кадров возлагается персональная ответственность за:

- организацию деятельности отдела по выполнению задач и функций, возложенных на отдел;

- организацию в отделе оперативной и качественной подготовки и исполнения документов, ведение делопроизводства в соответствии с действующими правилами и инструкциями;

- соблюдение сотрудниками отдела трудовой и производственной дисциплины;

- обеспечение сохранности имущества, находящегося в отделе, и соблюдение правил пожарной безопасности;

- подбор, расстановку и деятельность сотрудников отдела;

- соответствие действующему законодательству визируемых (подписываемых) им проектов приказов, инструкций, положений, постановлений и других документов.

Анализ существующих программных решений

В настоящее время существует множество готовых решений для автоматизации работы отдела кадров: подмножество систем, разработанных на базе 1С:Предприятие, а также прочих программных продуктов. Рассмотрим более подробно некоторые из них.

Программа кадрового обеспечения и делопроизводства «Кадры SB» от компании «Софт СБ» обеспечивает:

- ведение личных карточек сотрудников;

- учет сведений о работе;

- учет состава семьи;

- расчет выслуги лет;

- учет отпусков, отгулов, назначений и перемещений, поощрения, нарушения;

- учет образования, послевузовского образования, повышения квалификации, аттестации, профессиональной переподготовки;
- учет льгот, материальная помощь.

Прочие возможности программы:

- неограниченное число уровней подразделений и отделов (можно использовать одну базу данных для головного предприятия и его филиалов);

- учет вакансий и претендентов на должности;

- учет листов нетрудоспособности (больничных листов). Формирование любых отчетов (реестров) по больничным листам в MS Excel;

- формирование приказов и записок соответствующих форм (MS Word и Excel);

- формирование любых отчетов по персоналу. (MS Word и Excel);

- возможность самостоятельно формировать любые отчеты при помощи SQL-запросов;

- возможность использования в программе КЛАДР (Классификатор адресов России);

- возможность экспорта базы данных в другие форматы БД;

- возможность работы нескольких пользователей одновременно (сетевая версия);

- возможность ведения баз данных нескольких юридических лиц с одного рабочего места (сетевая версия);

- управление правами пользователей для работы в данной программе;

- учет трудовых книг и расчет любых видов стажей;

- штатное расписание. Возможность формирования штатного расписания отдельно по отделам, подразделениям (Excel);

- интеллектуальный мастер устройства на работу, перевода сотрудника на другую должность и увольнения сотрудника;

- внешний интерфейс с индивидуальными настройками;

- будильник, дни рождения, органайзер и многое другое.

Другой пример информационной системы учета кадров на предприятии - программа «Кадры» от компании «БухСофт» - одна из самых простых, по утверждению разработчиков, кадровых программ. Это интуитивно понятная условно бесплатная программа для ведения кадрового учета на предприятии со всеми необходимыми возможностями. Программа «Кадры» - составная часть комплексной программы «БухСофт: Зарплата, Табель, Кадры», но может быть приобретена отдельно от остальных функций.

Как и многие другие кадровые программы, программа "Кадры" от БухСофт позволяет вести учет таких данных как:

- персональные данные - фамилия, имя, отчество, дата и место рождения, пол, адрес регистрации и фактический адрес местожительства, сведения о документе, удостоверяющем личность, предусмотрена возможность вставки фотографии работника;

- налогообложение - заносятся данные, учитываемые при налогообложении доходов работника и формирования отчетов: ИНН, Код ИМНС (формируется автоматически на основании первых цифр ИНН), сведения об инвалидности, предоставление стандартных вычетов, учет доходов с прежнего места работы, сведения об инвалидности, стандартные вычеты, полученные доходы с предыдущего места работы;

- долги - в данной закладке заполняются начисленные, но не выплаченные доходы прошлых лет, долг налогоплательщика по налогу на доходы на начало года;

- начисления - заносятся данные о постоянных начислениях и удержаниях, на основе которых производится автоматическое начисление зарплаты;

- прочее - сведения о семейном положении работника, воинский учет;

- кадровые данные - заносятся сведения, относящиеся к рабочей деятельности сотрудника: табельный номер, статус, место работы, условия труда, характер работы, стаж, сведения о трудовом договоре, категории плательщика страховых взносов и номер страхового свидетельства в ПФР;

- перемещения - программа обеспечивает учет кадровых перемещений (прием на работу, переводы, увольнение). Информация вводится в отдельные поля, а общий наглядный отчет об имеющихся перемещениях выводится в виде таблицы;

- больничные - возможность заведения сведений о периодах болезни сотрудника с последующим автоматическим заполнением табеля рабочего времени за соответствующий период болезни и автоматическим расчетом «больничных» после выхода на работу с помощью функции «Постоянные начисления» в Расчете зарплаты;

- отпуск - возможность составления графика отпусков;

- командировки - программа обеспечивает учет сведений о командировках работника;

- образование - сведения о виде образования и учебном заведении работника, информация о послевузовском образовании, аттестация, квалификация, переподготовка, знание иностранных языков, профессия работника и т.д.

- сведения о поощрениях (наградах), льготах, на которые имеет право работник, возможность ввода любых дополнительных сведений.

Выбор и обоснование метода решения задачи

Так или иначе, всем организациям приходится сталкиваться с задачей учета кадров. Решить эту проблему можно по-разному.

Вариант первый - использовать для учета средства офисных приложений. Например, вести таблицу в MS Excel или другом табличном редакторе. Плюсы: тривиальное решение, требующее только навыков в использовании табличного редактора. Минусы: необходимость вводить все данные вручную, трудности при обновлении, отслеживании изменений в личных карточках работников.

Несколько лучше обстоят дела, когда вместо табличного редактора используется программа на базе популярной системы управления базами данных, например 1С. В этом случае значительно облегчен поиск данных, производится проверка дублирования записей, имеются средства построения различных отчетов. Но в целом эта система не избавлена от главного недостатка - необходимости вручную собирать информацию.

Второй вариант - воспользоваться специализированной информационной системой (ИС) для автоматизации учета кадров. Хранение информации в формате базы данных значительно облегчает последующее построение отчетов. Автоматическое обновление информации позволяет повысить оперативность при работе с базой данных.

ИС обычно приобретаются на достаточно долгий срок (среднее время «жизни» ИС - около 10 лет, но это не предел - во многих компаниях используются системы с гораздо большим «стажем» работы, правда, и обрастающими за это время новыми возможностями). Чтобы система автоматизации приносила ожидаемый эффект, она должна соответствовать данному предприятию - его возможностям, уровню развития и т.д. Стоимость ИС для небольшой фирмы не так уж и мала.

Несмотря на то, что в настоящее время существует множество готовых решений в области автоматизации учета кадров, в этих системах реализованы также десятки других задач, которые не всегда найдут свое применение на конкретном предприятии в силу специфики его деятельности.

Критериев выбора систем автоматизации, как и многих других, достаточно сложных и дорогих товаров (например, автомобилей), существует, конечно же, много. Какие-то из них крайне важны, какие-то могут отражать очень индивидуальные потребности. В подобных ситуациях следует во многом ориентироваться на «здравый смысл», а также иметь в виду некоторые

ключевые моменты, носящие специальный характер.

Выбирая систему автоматизации, стоит обратить внимание на следующее:

- что система автоматизации может делать, или какова ее функциональность;

- во что обойдется приобретение системы, запуск ее в эксплуатацию и поддержание в рабочем состоянии, т.е. какова ее совокупная стоимость владения (крайне важно знать именно общую стоимость, а не просто цену программного обеспечения).

- есть ли гарантии успешного завершения проекта внедрения и полноценного ввода системы в эксплуатацию;

- что у системы «внутри» и, следовательно, насколько она надежна, долговечна, производительна, в конце концов, современна;

- какова эффективность и возможные сроки окупаемости системы;

- уровень и качество сервиса в послепродажный период;

- возможность сопровождать и развивать систему силами специалистов самой фирмы;

- каковы перспективы системы, будет ли она развиваться и поддерживаться поставщиком в будущем [1].

Очень важно сначала выявить реальные потребности организации.

Определить реальные потребности организации в автоматизации - дело не простое. Очень хорошо, если в организации разработан план развития на несколько лет вперед, в котором определена роль информационных технологий и описана последовательность создания корпоративной автоматизированной системы управления. Такой продуманный подход дает наибольшую отдачу, существенно снижает риск выбрать «не ту» ИС и избежать проблем так называемой «лоскутной» автоматизации. При этом в качестве первоочередных задач может рассматриваться автоматизация наиболее критичных на данном этапе видов деятельности («узких» мест, от которых существенно зависит жизнь фирмы) или наиболее трудоемких при обработке традиционным способом (среди последних - например, бухгалтерский и налоговый учет, бюджетирование, расчет зарплаты, и др.) [2].

Таким образом, для реализации задачи автоматизации рабочего места сотрудника отдела кадров организации выбран вариант реализации посредством разработки информационной системы «Учет кадров организации».

Технология создания информационных систем предъявляет особые требования к методикам реализации и программным инструментальным средствам. Реализацию проектов по созданию информационных систем

принято разбивать на стадии анализа (прежде чем создавать информационных систем, необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственного кодирования, тестирования и сопровождения.

Сущность структурного подхода к разработке информационных систем заключается в ее декомпозиции (разбиении) на автоматизируемые функции: система разбивается на функциональные подсистемы, которые в свою очередь делятся на подфункции, подразделяемые на задачи и так далее. Процесс разбиения продолжается вплоть до конкретных процедур. При этом автоматизируемая система сохраняет целостное представление, в котором все составляющие компоненты взаимосвязаны. Основные этапы, на которые разбивается процесс проектирования информационной системы, следующие:

- концептуальное проектирование - сбор, анализ и редактирование требований к данным (обследование предметной области, изучение ее информационной структуры, выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами, моделирование и интеграция всех представлений);

- логическое проектирование - преобразование требований к данным в структуры данных. На выходе получаем СУБД-ориентированную структуру базы данных и спецификации прикладных программ [3-6];

- физическое проектирование - определение особенностей хранения данных, методов доступа и т.д. [7-10].

Современные объектно-ориентированные CASE-средства позволяют эффективно решать задачи проектирования приложений. Среди таких пакетов - Rational Rose, Together Control Center, BPWin, ERWin, Model Mart, Silverrun Business Process Modeller, Process Analyst.

Для разработки функциональной модели использовалось CASE-средство Computer Associates BPwin. BPwin является мощным инструментом для создания моделей, позволяющих анализировать, документировать и планировать изменения сложных бизнес-процессов. BPwin предлагает средство для сбора всей необходимой информации о работе предприятия и графического изображения этой информации в виде целостной и непротиворечивой модели. BPwin поддерживает три методологии: IDEF0, DFD и IDEF3, позволяющие анализировать ваш бизнес с трех ключевых точек зрения:

- с точки зрения функциональности

системы. В рамках методологии IDEF0 бизнес-процесс представляется в виде набора элементов-работ, которые взаимодействуют между собой, а также показывается информационные, людские и производственные ресурсы, потребляемые каждой работой;

- с точки зрения потоков информации (документооборота) в системе. Диаграммы DFD могут дополнить то, что уже отражено в модели IDEF3, поскольку они описывают потоки данных, позволяя проследить, каким образом происходит обмен информацией между бизнес-функциями внутри системы. В тоже время диаграммы DFD оставляют без внимания взаимодействие между бизнес-функциями;

- с точки зрения последовательности выполняемых работ: еще более точную картину можно получить, дополнив модель диаграммами IDEF3. Этот метод привлекает внимание к очередности выполнения событий.

Функциональное проектирование системы

В результате анализа предметной области была разработана функциональная модель БД для информационной системы. Проектирование проводилось на основе методологий IDEF0 и DFD.

Контекстная диаграмма, приведенная на рисунке 1, реализована с помощью методологии IDEF0.

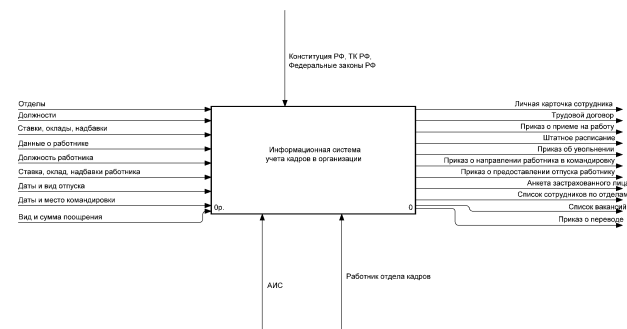


Рисунок 1 – Контекстная диаграмма системы

Входной информацией для системы являются отделы, должности, ставки, оклады и т.д. Ввод входной информации осуществляется работником отдела кадров.

Выходной информацией для системы являются выходные документы.

Функциональная декомпозиция системы, приведенная на рисунке 2, проводится на основе методологии IDEF0.

На этом уровне выполняются следующие функции:

- подготовка штатного расписания;
- определение вакансий;

– прием работника;
– выполнение операций с работником в ходе текущей работы сотрудника отдела кадров.

Функциональная декомпозиция активности «Прием работника», приведенная на рисунке 3, проводится на основе методологии DFD.

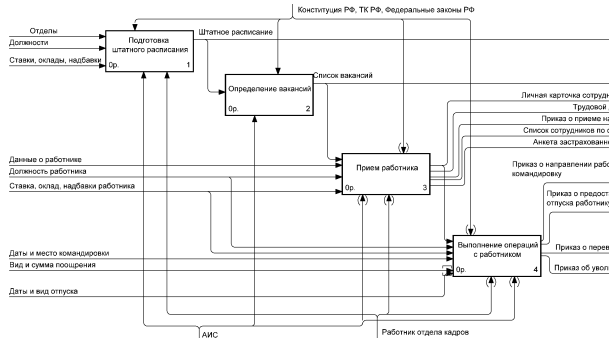


Рисунок 2 – Диаграмма декомпозиции системы

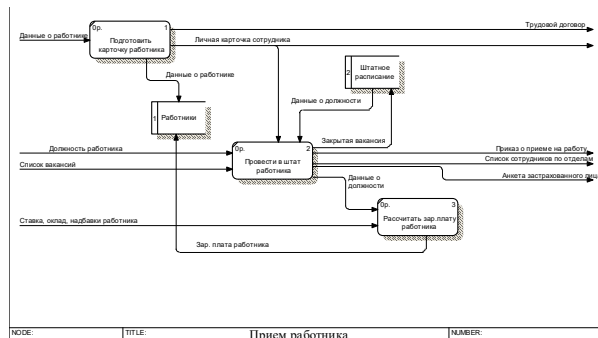


Рисунок 3 – Диаграмма декомпозиции деятельности «Прием работника»

Функциональная декомпозиция активности «Выполнение операций с работником», приведенная на рисунке 4, проводится на основе методологии DFD.

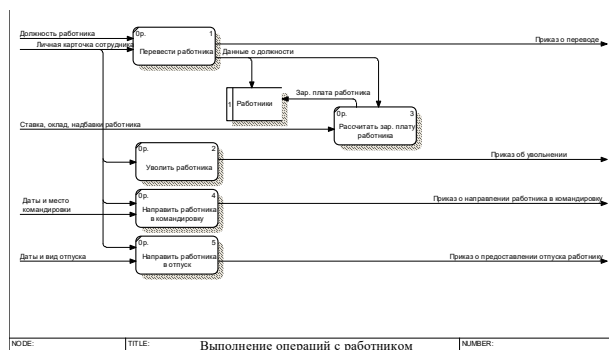


Рисунок 4 – Диаграмма декомпозиции деятельности «Выполнение операций с работником»

Экранная форма разработанной информационной системы приведена на рисунке 5.

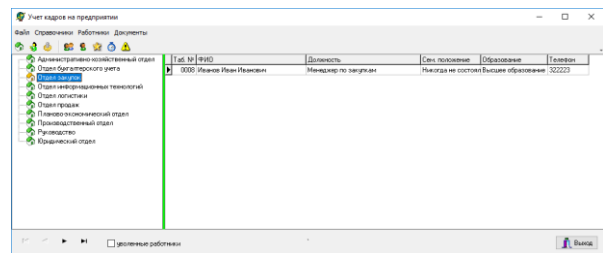


Рисунок 5 – Экранная форма информационной системы

Выводы

В результате проведенной работы была разработана информационная система учета кадров в организации, автоматизирующая работу работников отдела кадров.

Была проведена следующая работа:

- описана существующая система обработки информации в отделе кадров организации;
 - представлены обоснование выбора обеспечивающих технологий и проектных решений по программному и информационному обеспечению;
 - продуман пользовательский интерфейс, обозначены возможные пути расширения системы;
 - приведена информационная модель и представлено ее описание.
- В результате было спроектировано и разработано АРМ работника отдела кадров, реализованы следующие функции:
- подготовка и редактирование штатного расписания;
 - учет вакансий;
 - ведение личных карточек работников;
 - прием сотрудника на работу, перевод работника на новую должность;
 - ведение истории работы для каждого работника;
 - учет отпусков и командировок работников;
 - хранение в архиве базы данных информации об уволенных сотрудниках;
 - расчет заработной платы для каждого сотрудника с учетом надбавок и ставок;
 - подготовка данных для пенсионного фонда;
 - автоматизированная подготовка документов.

Литература

1. Боровская М.А., Катаев Б.В., Костюк А.И., Масыч М.А., Федосова Т.В. Технические

решения по оценке коммерческой эффективности инвестиционно-инновационных проектов в интерактивной среде// Инновационные технологии в экономике и управлении (статья). – Таганрог: Изд-во ТТИ ЮФУ, 2010. - № 9(10). -С. 183-196.

2. Боровская М.А., Морозова Т.В., Масыч М.А., Федосова Т.В., Шевченко И.К., Катаев Б.В., Кокорев А.А., Костюк А.И., Курунина Е.Н., Павлов А.Ю. Инструментарно-методическое обеспечение механизма взаимодействия инновационно-ориентированных субъектов на региональном уровне. Монография. / Под ред. Проф. М.А. Боровской. – Ростов-на-Дону - Таганрог: Изд-во ТТИ ЮФУ, 2010. –157 с.

3. А.И. Костюк. «Анализ методов типизации предметных областей баз данных». Материалы Международного конгресса по интеллектуальным системам и информационным технологиям (IS&IT'10). Таганрог, Изд-во ТТИ ЮФУ, 2010г.- С. 58-62.

4. В.Ф. Гузик, А.И. Костюк, Е.В. Ляпунцова, Б.В. Катаев, М.Ю. Поленов. Информационные технологии: Учебное пособие. – Таганрог: Изд-во ТТИ ЮФУ, 2013 – 115 с.

5. В.Ф. Гузик, Е. В. Ляпунцова, А.И.

Костюк, Б.В. Катаев. Базы данных: Учебное пособие. – Таганрог: Изд-во ТТИ ЮФУ, 2007. – 128 с.

6. В.Ф. Гузик, А.И. Костюк, Е.В. Ляпунцова. Базы данных. Руководство по выполнению лабораторных работ: Учебное пособие. - Таганрог: Изд-во Технологического института ЮФУ, 2010. – 101 с.

7. А. Д. Хомоненко, В. Э. Гофман, Е. В. Мещеряков. Delphi 7 – 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2010. – 1136 с.

8. Советов Б. Я., Цехановский В. В., Чертовский В. Д. Базы данных: теория и практика : учебник для студ. вузов. - 2-е изд.. - М. : Юрайт, 2012. - 463 с.

9. Култыгин О. П. Администрирование баз данных. СУБД MS SQL Server : учеб. пособие. - М. : Московская финансово-промышленная академия, 2012. - 228 с.

10. Баженова И. Ю. Основы проектирования приложений баз данных: учеб. пособие. - М. : БИНОМ. Лаборатория знаний : Интернет-Университет Информационных Технологий, 2006. - 324 с.

Костюк А.И., Лебедева Е.О. База данных учета кадров организации. Предложена информационная система учета кадров организации, автоматизирующая работу работников отдела кадров, разработанная на основе контекстной диаграммы, диаграммы декомпозиции, диаграмм логической и физической моделей базы данных Разработанная база данных позволяет повысить эффективность автоматизации документооборота за счет существенного снижения времени, необходимого для обработки информации и оформления кадровых документов.

Ключевые слова: автоматизированное рабочее место, информационная система, база данных, отдел кадров, программное обеспечение.

Kostyuk A.I., Lebedeva E.O. Information system for accounting staff of the organization. A personnel record information system organization, automating the work of the personnel department employees, developed on the basis of contextual diagrams, decomposition diagrams, logical and physical database model diagrams of data developed database allows to increase the efficiency of workflow due to a significant reduction in the time required for data processing and clearance of personnel documents.

Keywords: workstation, database, information system, human resources software.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.056.5

Программная реализация алгоритмов обфускации программного кода языка JavaScript

С.В. Медгаус, А.В. Чернышова

Донецкий национальный технический университет
medgaus-sergey@yandex.ru, alla@donntu.org

Медгаус С.В., Чернышова А.В. Программная реализация алгоритмов обфускации программного кода языка JavaScript. В тексте данной статьи представлен краткий обзор существующих обфускаторов для языка JavaScript. После их анализа предложено реализовать собственный обфускатор, применяющий сложные преобразования исходного кода. В статье подробно описаны применяющиеся алгоритмы запутывания программного кода языка JavaScript.

Введение

На данный момент проблема сохранения прав собственности на исходный код достаточно актуальна. В отличие от настольных приложений, в которых существуют различные средства защиты и лицензирования, в web-приложениях такие способы защиты не применимы и весь исходный код скриптов доступен для просмотра любому желающему через браузер.

Подробный анализ существующих способов защиты web-приложений рассмотрен в [1]. Исходя из изложенного в статье материала, можно сказать, что обфускация – это наименее ресурсоёмкий способ защиты исходного кода web-приложений. «Обфускация (от лат. obfuscare — затенять, затемнять; и англ. obfuscate — делать неочевидным, запутанным, сбивать с толку) или запутывание кода — приведение исходного текста или исполняемого кода программы к виду, сохраняющему его функциональность, но затрудняющему анализ и понимание алгоритмов работы. Существуют специальные программы, производящие обфускацию, так называемые обфускаторы» [2].

На текущий момент существуют различные программные продукты, которые предлагают обфускацию исходного кода, однако эффективные алгоритмы обфускации предлагают только платные продукты, бесплатные же работают на уровне минификации кода (уменьшение размера исходного кода, путём сокращения имён переменных и функций, удаление символов форматирования кода). В работе [1] были рассмотрены и проанализированы такие полнофункциональные и эффективные программные продукты как YUI Compressor [3], Packer [4], JavaScript Obfuscator [5] и Google Closure Compiler [6]. После анализа существующих обфускаторов было принято

решение разработать комбинацию эффективных обфусцирующих алгоритмов для защиты программного кода языка JavaScript.

Целью работы являются разработка и программная реализация алгоритмов обфускации для языка JavaScript с последующим объединением их в комплексный алгоритм обфусцирующего преобразования и использование его при создании обфускатора.

1 Проектирование программного продукта

После анализа существующих обфускаторов для языка JavaScript, была разработана диаграмма прецедентов для проектируемого обфускатора (см. рис. 1).

Диаграмма прецедентов UML – это диаграмма, показывающая связи между актёрами и действиями (прецедентами), которые они могут выполнять.[7].

Как видно из рис. 1, пользователь может загружать файл с исходным кодом, сохранять обфусцированный код в файл, запускать обфускацию, а также выбирать режимы обфускации:

- удаление форматирования;
- преобразование условных конструкций;
- логическое преобразование;
- сокращение констант;
- кодирование числовых констант;
- кодирование строковых констант;
- переименование переменных.

Все вышеперечисленные режимы обфускации применяются в заданном программой порядке, который невозможно изменить, то есть являются этапами. Это позволяет максимально эффективно запутать исходный код, так как появляются связи ещё и между самими режимами обфускации.

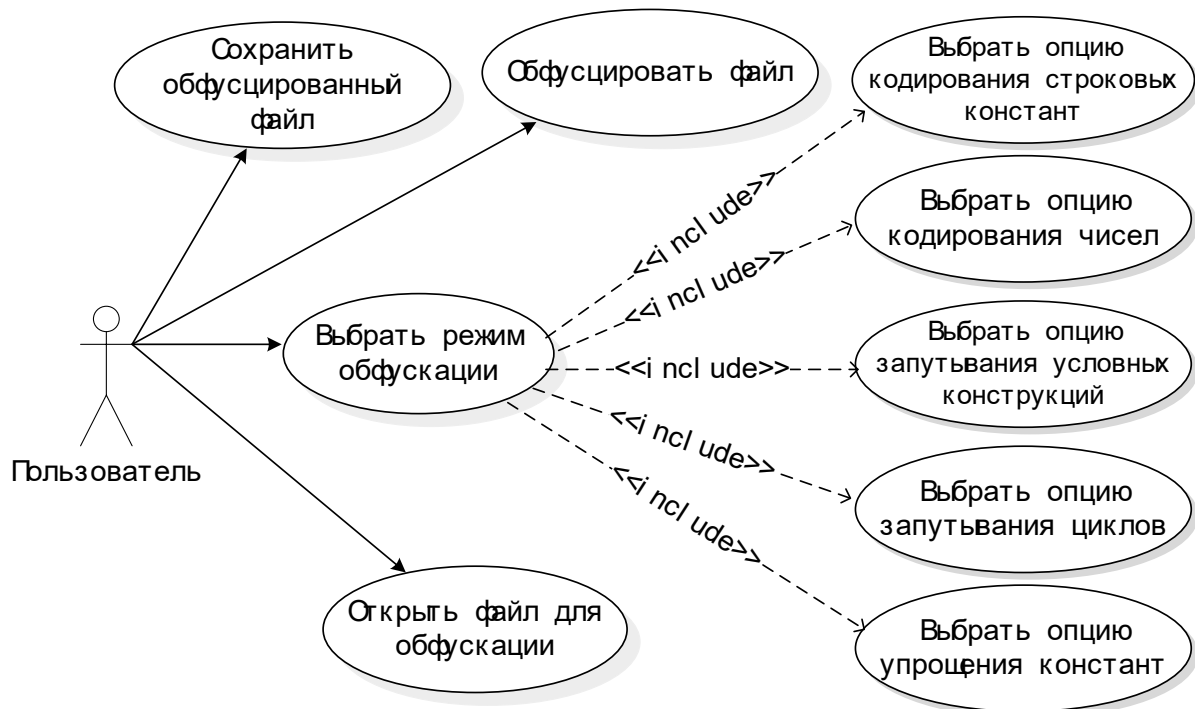


Рисунок 1 – Диаграмма вариантов использования обфускатора

Исходя из спроектированной диаграммы вариантов использования, была разработана диаграмма потоков данных обфускатора, которая представлена на рис. 2.

Приложение будет состоять из одной формы, в которой будет содержаться главный рабочий класс Обфускатор, который будет

выполнять всю работу. В свою очередь, в зависимости от выбранных режимов, обфускатор будет использовать различные запутыватели, которые предоставляет программа. После обфускации программа будет выдавать выходной файл, который будет содержать обфусцированный JavaScript код.

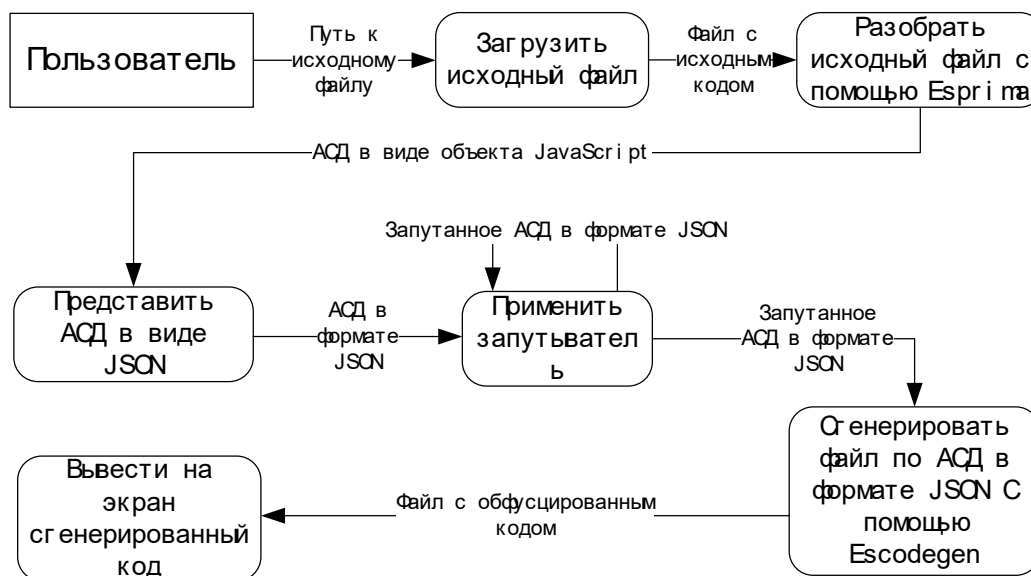


Рисунок 2 – Диаграмма потоков данных работы обфусцирующей программы

Все остальные диаграммы, разработанные при проектировании обфускатора, представлены в [8].

2 Алгоритмы запутывающих преобразований

Эффективность программы-обфускатора зависит от её алгоритмов, которые преобразовывают (запутывают) исходный код программ [9].

2.1 Обобщённый алгоритм работы обфускатора

Необходимо отметить, что основным объединяющим алгоритмом программы является алгоритм, в котором объединяются все запутывающие алгоритмы и используются для приведения исходного кода в непонятный для человека вид (см. рис. 3).

```
подсчитать количество объектов-запутывателей;  
подключить движок Nashorn;  
распознать код с помощью Esprima;  
перевести из формата Object в JSON с  
    помощью встроенной библиотеки JSON;  
распознать с помощью JSONSimple дерево в формате JSON;  
  
FOR (для каждого запутывателя) {  
    получить текстовое представление этапа;  
    установить прогресс в зависимости  
        от номера запутывателя;  
    оповестить текстовым представлением;  
    запутать код;  
}  
  
установить прогресс на последний этап;  
опубликовать этап "Построение кода";  
  
вызвать метод "Запутать" у объекта  
    класса ПеремешивательФункций;  
представить АСД в формате Object;  
задать опции генерации кода, учитывая форматирование;  
сгенерировать код с помощью библиотеки Ecodegen;  
оповестить об окончании обфускации;  
  
вернуть обфусцированный код;
```

Рисунок 3 – Обобщённый алгоритм работы программы

2.2 Алгоритм преобразования выражений условного оператора if-else

Суть этого преобразования заключается в том, чтобы преобразовать допустимые выражения оператора if-else в тернарный оператор. Этот алгоритм нацелен на ухудшение читаемости кода, без каких-либо фундаментальных преобразований. Алгоритм преобразования на псевдоязыке представлен на рис. 4.

2.3 Алгоритм логического преобразования

Данный алгоритм представляет собой простое инвертирование сравнений таким

образом, чтобы вместо исходного оператора использовалось отрицание противоположного оператора с отрицаемыми операндами (см. рис. 5) [6]. На схеме алгоритма указана только часть логических операторов. Операторы, не попавшие на схему, преобразуются по такому же алгоритму.

Необходимо отметить, что можно использовать и другие логические преобразования, применяя логические эквиваленции, однако в JavaScript результатом логического выражения может быть не только true или false, но и ещё и объект, поэтому использование более сложных формул преобразований может приводить к ошибкам.

```

выполнить проход по АСД (
    IF (есть условные конструкции с пустой веткой else) THEN
        устанавливаем флаг необходимости доп. функций;
)

IF (флаг необходимости доп. функций) {
    добавляем в массив функций случайное кол-во
    случайно сгенерированных функций;
    FOR (каждой функции) {
        сгенерировать случайное арифметическое выражение
        из нескольких арифметических операндов;
        представить выражение в виде строки;
        внести в тело функции в виде получения кода из
        символьных кодов;
        занести тело функций в конец исходного кода программы;
    }
}

выполнить проход по АСД (
    IF (текущий узел = "условная конструкция" И
        (выражение не содержит вызовов методов объектов
        ИЛИ системных вызовов)) THEN {
        IF (ветка else пустая) {
            занести туда вызов сгенерированной функции;
        }
        поменять местами ветки if и else;
        поменять тип узла на "тернарный оператор";
    }
)

```

Рисунок 4 – Алгоритм преобразования выражений условного оператора if-else

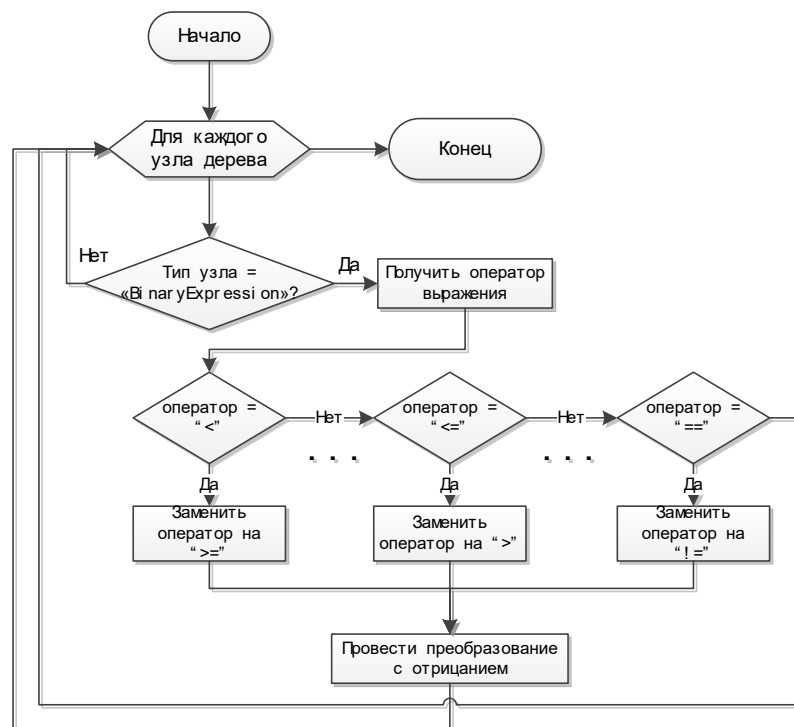


Рисунок 5 – Алгоритм преобразования логических выражений

2.4 Алгоритм сокращения констант

По стандарту ECMA Script 5.0 в языке JavaScript ещё нет констант, которые бы обеспечивал сам язык, поэтому программа сама анализирует переменные, которые не изменяют своего значения в исходном коде пользовательской программы.

Данное преобразование состоит из следующих этапов:

- найти возможные константы в коде;
- исключить меняющиеся переменные;
- исключить переменные, чьё значение не является простым, то есть не является числом, строкой или булевым значением;
- заменить в местах использования констант их значением.

Детальное описание алгоритма представлено на рис. 6.

```

FOR (для каждого узла дерева) {
  IF (узел - это объявление переменной) THEN {
    получить имя переменной;
    поместить в кеш-таблицу имя переменной и её значение;
  }
}
FOR (для каждого узла дерева) {
  IF (узел используется с оператором
    инкремента или декремента) THEN {
    удалить переменную из кеш-таблицы;
  }
  IF (узел - присвоение переменной значения) THEN {
    IF (переменная неинициализированная) THEN {
      присвоить новое значение и занести в кеш-таблицу;
    } ELSE {
      удалить переменную из кеш-таблицы;
    }
  }
}
FOR (для каждой переменной в кеш-таблице) {
  IF (инициализированное значение не простое) THEN {
    удалить переменную из кеш-таблицы;
  }
}
FOR (для каждого узла дерева) {
  IF (используется переменная, которая есть
    в кеш-таблице) THEN {
    заменить использование переменной её значением;
  }
}
FOR (для каждого узла дерева) {
  IF (объявление переменной, которая есть в кеш-таблице) THEN {
    удалить объявление переменной;
  }
}

```

Рисунок 6 – Алгоритм сокращения констант

2.5 Алгоритм кодирования чисел

Данный алгоритм очень важен для программной системы, так как большая часть исходного кода в любом языке программирования использует огромное количество чисел для счётчиков различного рода, для циклов и для числовых констант.

Суть этого алгоритма заключается в том, чтобы заменить числа некоторым набором арифметических выражений, и после этого представить все числа в 16-ичной системе счисления [6]. Детальный алгоритм представлен на рис. 7.



Рисунок 7 – Алгоритм кодирования чисел

2.6 Алгоритм кодирования строк

Данный алгоритм также является важным в работе программной системы. Почти во всех программах используются какие-либо строковые константы, будь-то сообщения об ошибках или текстовые элементы интерфейса. Для того, чтобы труднее было разобрать исходный код программы, необходимо преобразовать и скрыть явное использование таких строковых констант.

В данном преобразовании строки представляются в виде конкатенации вызовов различных функций, а также в виде кодировки BASE64. Всё разбиение строковых констант происходит каждый раз случайно, обеспечивая различный исходный код на выходе обфускатора [7].

Детальный алгоритм представлен на рис. 8.

2.7 Алгоритм переименования переменных

Данное преобразование исходного кода является самым не затратным с точки зрения ресурсов преобразования, но с точки зрения конечного запутывания является одним из самых эффективных алгоритмов обфускации.

Главная особенность этого алгоритма в том, чтобы представить имена переменных и функций в виде непонятного для человека набора символов [6]. Подробный разбор алгоритма представлен на рис. 9.

```

FOR (для каждого узла дерева) {
    случайно получить флаг "необходимо ли делить слово";
    IF (узел - это строковый литерал И
        ...
        есть флаг "делить слово") THEN {
        получить длину строки;
        получить случайное число до длины строки;
        разбить слово на два по случайному числу;
    }
}
FOR (для каждого узла дерева) {
    IF (узел - это строковый литерал) THEN {
        занести каждую строковую константу в массив констант;
    }
}
FOR (для каждого узла дерева) {
    случайно получить флаг "необходимо
    ...
    ли выносить слово в отдельную функцию";
    IF (узел - это строковый литерал И
        ...
        есть флаг "выносить слово") THEN {
        заменить использование строковой константы на
        вызов отдельной функции;
        установить флаг "отдельная функция использовалась";
    }
}
IF ("отдельная функция использовалась") {
    занести в конец исходного кода тело отдельной функции
    для строковых констант;
}
FOR (для каждого узла дерева) {
    случайно получить флаг "необходимо ли преобразовать в Base64";
    IF (узел - это строковый литерал И
        ...
        есть флаг "необходимо преобразовать") THEN {
        заменить использование строковой константы на
        закодированное значение в Base64 и вызов функции
        декодирования;
    }
}
}

```

Рисунок 8 – Алгоритм кодирования строковых констант

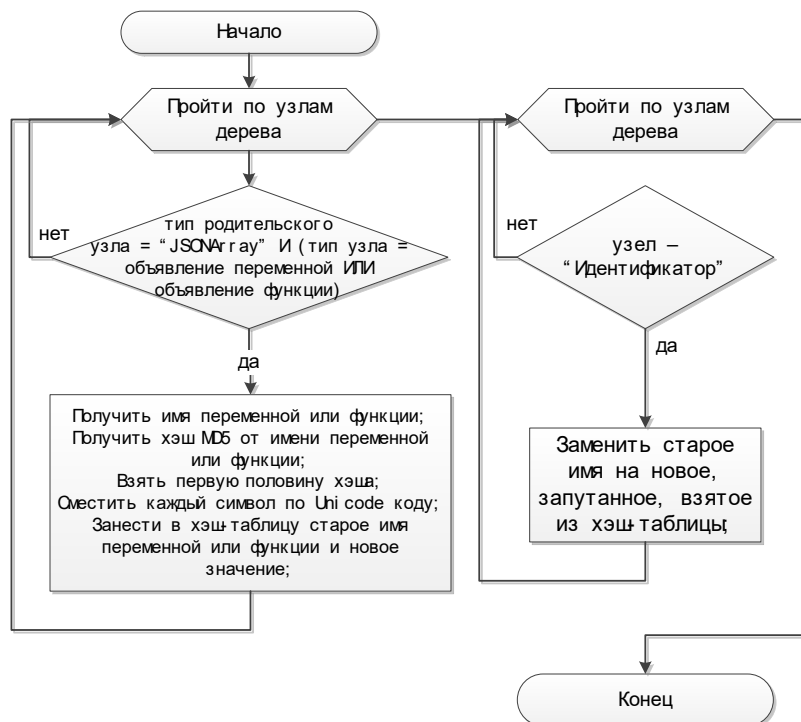


Рисунок 9 – Алгоритм переименования переменных

2.8 Алгоритм переставления функций

Данный алгоритм не является напрямую запутывающим преобразованием, его нельзя выбрать через интерфейс пользователя. Он служит для того, чтобы перемешивать глобальные функции, которые могли появиться вследствие других запутывающих преобразований (мусорные

функции). Иначе, если это преобразование не использовать, то мусорные функции располагались бы в конце документа с исходным кодом и могли бы легко быть удалены человеком, который взялся бы деобфусцировать исходный код. Детальное описание алгоритма представлено на рис. 10.

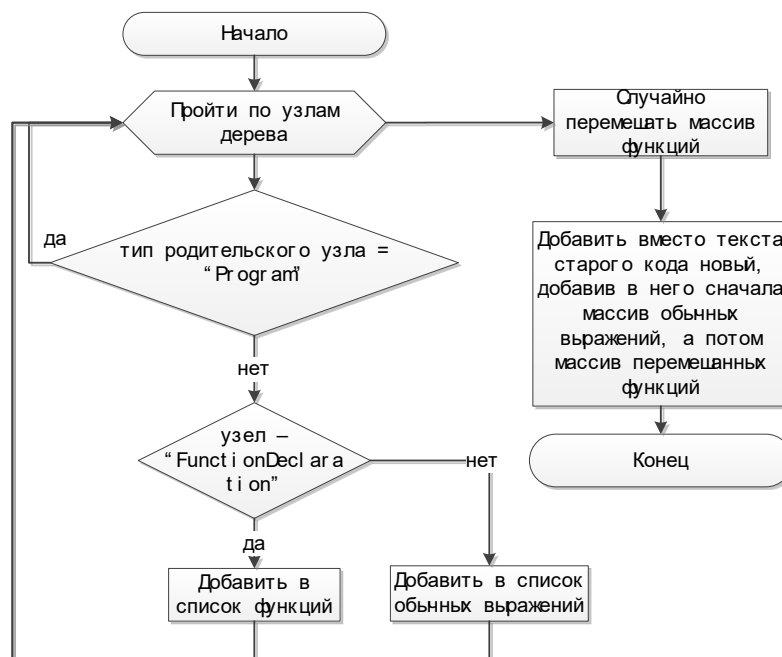


Рисунок 10 – Алгоритм переставления функций

3 Результат разработки алгоритмов и их реализации в программном продукте

После разработки алгоритмов, они были реализованы на языке Java, где и были объединены в один главный обфусцирующий алгоритм.

На рис. 11 представлена экранная форма обфускатора, на которой видно, что пользователь выбрал все доступные режимы обфускации. Также видно, что исходный код существенно изменился.

Как показало тестирование на экспертных группах, код стал более запутанным. Эксперты сделали субъективное заключение – сложность понимания исходного кода увеличилась в 3 раза.

Выводы

В ходе выполнения данной работы были проанализированы существующие обфускаторы

языка JavaScript. После их детального изучения, были выявлены присущие им недостатки, учитывая которые, было решено разработать и реализовать собственные эффективные алгоритмы обфускации программного кода языка JavaScript.

Также была разработана последовательность объединения запутывающих преобразований (общий алгоритм работы обфускатора) с максимальной степенью запутанности.

Дальнейшие разработка и реализация алгоритмов обфускации являются перспективными, так как разработанные алгоритмы работают напрямую с кодом, меняя внешнее представление кода, в то время как, преобразования потока управления программ не производится. То есть разработанные алгоритмы принадлежат к категориям лёгкой и средней степеням обфускации, а алгоритмы обфускации сложной степени необходимо разрабатывать.

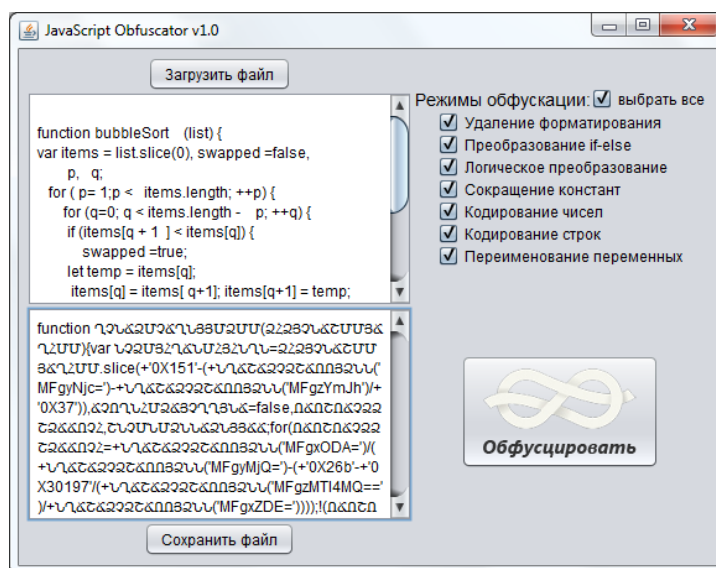


Рисунок 11 – Экранная форма обфускатора

Литература

1. Обзор существующих обфускаторов и их алгоритмов. Компьютерная и программная инженерия - 2015 год: - Донецк: ДонНТУ, 2015. – С.117-119.
2. Википедия. Обфускация [электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Обфускация>.
3. YUI Compressor [электронный ресурс]. – Режим доступа: <https://github.com/yui/yuicompressor>.
4. Packer [электронный ресурс]. – Режим доступа: <http://dean.edwards.name/packer>.
5. JavaScriptObfuscator. Canada [электронный ресурс]. – Режим доступа: <https://www.javascriptobfuscator.com/>.
6. Google Closure Compiler [электронный ресурс]. – Режим доступа:

<https://developers.google.com/closure/compiler/>.

7. Диаграммы прецедентов: крупным планом [электронный ресурс]. – Режим доступа: <http://www.intuit.ru/studies/courses/1007/229/lecture/5962>.
8. Проектирование обфускатора для языка JavaScript. ИУСМКМ - 2016: VII Международная научно-техническая конференция, 26 мая 2016: - Донецк: ДонНТУ, 2016. – С. 167-173.
9. LynX. Обфускация и защита программных продуктов // CITForum. [электронный ресурс]. – Режим доступа: <http://citforum.ru/security/articles/obfus>.
10. Java Obfuscator – String Encryption // zelix.com [электронный ресурс]. – Режим доступа: <http://www.zelix.com/klassmaster/featuresStringEncryption.html>.

Медгаус С.В., Чернышова А.В. Программная реализация алгоритмов обфускации программного кода языка JavaScript. В тексте данной статьи представлен краткий обзор существующих обфускаторов для языка JavaScript. После их анализа, предложено реализовать собственный обфускатор, применяющий сложные преобразования исходного кода. В статье подробно описаны применяющиеся алгоритмы запутывания.

Ключевые слова: обфускатор, Java, JavaScript, запутывание, обфускация, алгоритмы обфускации, Esprima, Nashorn

Medgaus Sergey, Chernyshova Alla Program realization obfuscating algorithms of JavaScript source code. There is presented short overview of existing obfuscators for JavaScript language in the text of this article. It is proposed to realize own obfuscator that uses complex transformation of source code. There are described different applied mangling algorithms.

Keywords: obfuscator, Java, JavaScript, mangling, obfuscation, obfuscating algorithms, Esprima, Nashorn

Статья поступила в редакцию 20.11.2016

Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.942

Моделирование и информационные технологии при распознании лица человека по его мимическим фотографиям

А.С. Миненко

Донецкий национальный технический университет
sam_dntu@mail.ru

Миненко А.С. Информационные технологии при моделировании и распознании лица человека по его мимическим фотографиям. Работа посвящена разработке методов, моделей, алгоритмов и компьютерных средств, для моделирования распознавания эмоциональной составляющей фотографических изображений лица и трехмерных моделей человека.

Ключевые слова: аппроксимация, мимические проявления, эмоции, алгоритм, сплайн-аппроксимация.

Введение

Математические методы и компьютерные технологии анализа и синтеза лица человека, эмоции на нем исследуются и разрабатываются в ведущих научных организациях мира, в частности, в Массачусетском технологическом институте, Оксфордском, Московском, Санкт-Петербургском университетах, др. В Украине исследованием таких проблем занимаются в Институте кибернетики им. В.М.Глушкова НАН Украины, Киевском национальном университете имени Тараса Шевченко, ГОУВПО Донецкий национальный технический университет и других учреждениях.

Приведем обзор теорий, которые объясняют механизмы возникновения и проявления эмоций [1-5].

Формальная модель эмоций

Для формализации эмоций, чтобы избежать двусмысленностей при их феноменологическом описании, предлагается перейти к изучению ситуаций, в которых эти эмоции возникают [1]. То есть, при определении эмоций, в наиболее общем виде описывается ситуация, в которой они возникают. Различать будем эмоции и их обозначение. Под обозначением будем иметь в виду вектор (Em) (то есть абстрактное понятие) со следующими признаками:

$$Em_i^n = (\xi_1, \xi_2, \xi_3), i = 1 \dots 8,$$

где ξ – бинарный признак, который классифицирует эмоции:

ξ_1 – признак, который определяет знак эмоции – позитивная (1) эмоция или негативная (0). Будем называть эмоцию позитивной, если она возникает в связи с удовлетворением потребности или

достижением цели, и, следовательно, негативной – в связи с неудовлетворением или недостижением;

ξ_2 – признак, который определяет время возникновения эмоции относительно действия (предвидевшая (0) и констатирующая (1) эмоции). Предвидя эмоции возникают до действия связанного с достижением (недостижением) цели, предвидят ее;

ξ_3 – признак, который определяет направление эмоции. По этому признаку выделяют эмоции направленные на себя (1) и направленные на внешние объекты, на других людей (0). Сокращенная функция эмоции лежит в том, что эмоции подготавливают организм к определенному событию которое возникнет. Эмоции служат для решения определенных жизненных трудностей, сложных положений. Каждая эмоция готовит человека к какому-то событию. Это событие может происходить с внешним объектом или с самим человеком. Например, гнев нацелен на устранение преград для достижения цели, и, таким образом, направленный на внешний объект. Грусть готовит человека обходиться той целью, которую не удалось достичь, и направлена на себя.

Рассматривается задача моделирования мимических проявлений эмоций формального описания базовых эмоций. Для поиска пространства характеристических признаков, построения базиса этого пространства, воспроизведение производных эмоциональных состояний со следующим использованием выпуклой комбинации, предлагается следующее:

- создание множества фотографических изображений, на которых актерами воссоздаются ситуации (ξ_1, ξ_2, ξ_3) , у которых

возникают базовые эмоции, и описание мимики, свойственной этим эмоциям;

- анализ полученного множества с целью выявления областей, которые содержат характеристические признаки эмоций и описание их (используя анатомические признаки и методику Facial Action Coding System);
- создание в пространстве характеристических признаков базиса для следующего расписания по нему произвольных векторов мимических проявлений эмоциональных состояний (как выпуклой комбинации базисных эмоциональных состояний);
- анализ характеристических признаков и ранжирование их по степени влияния в рамках предложенного базиса.

Для анализа полученного множества фотографических изображений с целью выявления областей, которые содержат характеристические признаки эмоций, был использован подход, предложенный авторами методики FACS [1-6]. В ходе исследования, анализируя мимические проявления в разрезе зон лица, было получено 21 характеристический признак, комбинация которых образует базис мимических проявлений эмоций. То есть, мимические проявления эмоций (Em) были представлены в виде вектора:

$$Em_i^n = (\mu_1, \dots, \mu_{21}), i = 1 \dots 8,$$

где $\mu_j \in [0;1]$ – характеристический мимический признак (при $\mu = 0$ – нет признака, а при $\mu = 1$ влияние признака максимально).

Набор 8-ми векторов, указанных таким образом, создает базис (B_{ij} , где $i = 1, \dots, 21, j = 1, \dots, 8$) пространство мимических проявлений эмоциональных состояний.

Таким образом, произвольных вектор $b = (\mu_1, \dots, \mu_{21})$, полученных путем анализа изображений с каким либо эмоциональным состоянием, можно разложить по базису B и получить описание эмоции, как выпуклой комбинации:

$$x = (B^T B)^{-1} B^T b$$

где B – базисная матрица эмоциональных состояний; B^T – транспонирована матрица B ; b – вектор, который описывает мимическое проявление произвольного эмоционального состояния;

$$x = (a_1, \dots, a_8)$$

где a_i – коэффициенты выпуклой комбинации ($\sum a_i = 1, a \in [0;1]$) для каждой из 8-ми базовых эмоций.

Так как невозможно мимически определить источник происхождения эмоций (η), получаем следующее множество из 21-й эмоции, которую можно определить предложенными способами.

Использование контурных моделей на базе сплайн кривых для анализа мимических проявлений эмоций

Для анализа мимических проявлений эмоций, имея в виду формальную модель эмоций состояния человека, предлагается оригинальная технология. Основная ее идея заключается в том, чтобы предложить алгоритм автоматического получения гибких шаблонов контуров бровей, глаз и рта. Для построения этих шаблонов нужно предложить:

- метод который даёт возможность выделить наиболее важные части лица человека, а именно: нос, брови, глаза, губы и щеки.
- алгоритм получения точечных кривых, которые соответствуют контурам лица, которые будут анализироваться;
- алгоритм аппроксимации полученных точечных кривых с помощью сплайн кривых.

Определение опорных точек рта

Для воспроизведения изображения рта необходимо точно определить его положение. Крайние точки можно найти программными методами, но существуют модели голов, в которых распознать их с необходимой точностью достаточно сложно. Сделаем это в ручную, что не потребует много времени и не будет давать значительных погрешностей. Обозначим их соответственно (X_n, Y_n) – крайняя справа и (X_l, Y_l) – крайняя левая точки. С помощью этих двух точек будем определять все части лица.

Построение части профиля, которая проходит через нос

При построении профиля носа используем несколько особенностей, которые определяют его среди всех частей лица:

- со всех точек лица высшая точка по z находится на носу;
- при движении вдоль профиля от рта до носа мы наталкиваемся на

стремительный перепад высот, который определит начало профиля носа;

- нос переходит в лоб, причем, если отойти от профиля уменьшая или увеличивая абсциссу x , то наблюдается уменьшение координаты z , но когда мы доберемся лба, то координата z или не уменьшится, или это уменьшение будет незначительным.

Нахождение начальной точки. С начала возьмем точку, которая находится посередине рта ($\frac{x_n + x_{\text{л}}}{2}, \frac{y_n + y_{\text{л}}}{2}$). Отметим, что эта точка находится на профиле головы.

Нахождение начала профиля носа. Начнем движение от начальной точки по направлению увеличения ординаты y , проверяя при этом изменение координаты z . Таким образом, дойдем до некоторой точки, в окрестности которой начнет резко увеличиваться координата z (то есть $z_{yy} > z_{y21}$ - некоторое пороговое значение, которое определяется на практике; в разработанном программном продукте используется значение $z_{y21} = 2.5$). Эта точка и определит начало профиля носа.

Нахождение верхушки носа. Верхушку определим продолжая двигаться в направлении увеличения ординаты y пока $z'_y > 0$, то есть пока увеличивается координата z . Искомая точка находится по условиям $z'_y(x, y + \lambda) < 0$, $z'_y(x, y - \lambda) > 0$

Нахождение конца профиля носа.

Далее воспользуемся свойством носа. Начнем исследовать склоны с профилем, который определяется значениями

$$z'_x(x - \delta, y), z'_x(x + \delta, y). \quad \text{Условия}$$

$$z'_x(x - \delta, y) > z_{x11}, \quad z'_x(x - \delta, y) > z_{x11},$$

$$z'_x(x + \delta, y) < z_{x11} \quad (\text{где } z_{x11} - \text{некоторое}$$

пороговое значение, которое определяет тангенс угла наклона; $z_{x11} = 0.5$) характеризуют наличие склонов.

Двигаясь в направлении увеличения ординаты y доберемся до точки для которой $z'_x(x - \delta, y) > \varepsilon$,

$$z'_x(x + \delta, y) < \varepsilon, \quad z'_x(x - \delta, y + \lambda) < \varepsilon,$$

$$z'_x(x + \delta, y + \lambda) > \varepsilon. \text{ Точка } (x, y) \text{ есть искомой.}$$

Обозначим ее $(x_{\text{ня}}, y_{\text{ня}})$.

Построение границы носа

На предыдущем шаге был определен профиль носа, используя склоны на абсциссе x . Теперь используя их определим границу носа.

Начнем движение с точки $(x_{\text{ня}}, y_{\text{ня}})$, которая представляет собой точку перехода носа в лоб, вдоль профиля по направлению уменьшения ординаты y с шагом δ .

Рассмотрим k -й шаг. $x_k = x_{\text{ня}}, y_k = y_{\text{ня}} - k\lambda$

(x_k, y_k) - точка профиля.

Зафиксируем ординату y и начнем с шагом δ уменьшать абсциссу x , наблюдая за величиной $z'_x(x, y)$. Движение будет продолжаться пока $z_x(x, y)$ не станет меньше некоторого порогового значения z_{x12} ($z_{x12} = 0.6$). Так найдем некоторую точку $(x_k^{(n)}, y_k^{(n)})$, которая лежит на правой границе носа. Аналогично, увеличивая абсциссу x получаем соответствующую точку $(x_k^{(л)}, y_k^{(л)})$ на левой границе носа. Совокупность точек $(x_k^{(n)}, y_k^{(n)})$, $(x_k^{(л)}, y_k^{(л)})$ будет составлять границу носа, а все точки, которые лежат в этой области формируют сам нос.

Построение линии бровей

Линии бровей начинаются с переносицы. Эту точку было найдено при построении профиля носа и обозначено как точку перехода нос-лоб. Рассмотрим некоторые особенности бровей.

Пусть имеем точку (x, y) на бровях. Тогда при уменьшении ординаты y на λ произойдет резкое уменьшение ординаты z . Это условие запишем так: $z_y(x, y) < z_{y22}$, ($z_{y22} = 1.5$)

Начнем движение с точки (x, y) в направлении уменьшения абсциссы x по линии $y = y_n$. Рассмотрим k -й шаг.

Имеем точку (x_k, y_k) : $x_k = x_{\text{пл}} - k\delta$, $y_k = y_{\text{пл}}$.

Рассмотрим линию $X = x_k$.

Будем двигаться вдоль нее по точкам $(x_k, y_k + l\lambda)$, $l = 1, 2, \dots$ пока не найдем такую точку $(x_k^{(n)}, y_k^{(n)})$, что будет выполняться неравенство $z''_{yy}(x_k, y_k + l\lambda) < z_{y22}$.

Движение по направлению уменьшения абсциссы x останавливаем когда выполняется условие

$z_x''(x_k, y_k) > z_{y23}(z_{y23} = -1.3)$, то есть, когда будет наблюдается резкое уменьшение координаты z .

Аналогично строится и левая бровь.

Построение контура глаз

Рассмотрим построение правого глаза.

Для распознавания глаза используем найденную ранее бровь.

Нахождение крайней правой точки глаза. Начнем движение от средней точки брови по направлению уменьшения абсциссы x с шагом δ .

Рассмотрим k -й шаг. Найдем (x_k, y_k) - точка брови.

Зафиксируем абсциссу x и начнем двигаться по направлению уменьшения ординаты y :

$$x_k = x_{nl} - k\delta,$$

Точки контура глаза найдем из условий

$$z_{yy}''(x_k, y_{kl1}) > z_{y24}$$

$$z_{yy}''(x_k, y_{kl2}) < 0, l_1 < l_2 < l_3,$$

$$z_{yy}''(x_k, y_{kl3}) > z_{y24}$$

где z_{y24} - некоторое пороговое значение, которое определяется на практике. Эти условия определяют наличие вогнутости на краях глаза и выпуклости посередине. Если возникнет ситуация когда $l_3 - l_2 < 2$, то эта неравенство определит крайнюю правую точку глаза.

Обратное движение. Движение организуем аналогично первому этапу с отличием в том, что двигаться по брови будем справа налево пока не найдем крайнюю левую точку глаза.

Аналогично будем контур левого глаза.

Построение контура губ

Построение линии, которая проходит между губами. Движение начнем с крайней правой точки губ (x_n, y_n) (напоминаем, что эта точка задается вручную) по направлению увеличения абсциссы x с шагом δ .

$(x_{k(c)}, y_{k(c)})$ - точка на искомой линии

$$x_{k(c)} = x_n + k\delta$$

Величину $y_{k(c)}$ определим так:

$$y_{k(c)} = y_{k-1(c)} + \hat{l}\lambda, \quad (1)$$

Построение линии, которая ограничивает верхнюю губу. Движение начнем с крайней правой точки губ (x_n, y_n) по направлению увеличения абсциссы x с шагом δ .

$(x_{k(c)}, y_{k(c)})$ - точка на искомой линии

$$x_{k(c)} = x_n + k\delta.$$

Величину $y_{k(c)}$ определим следующим образом:

$$y_{k(c)} = y_{k(c)} + \hat{l}\lambda, \quad (2)$$

Построение линии, которая ограничивает нижнюю губу. Эта линия строится аналогично предыдущей с отличием в обозначении $\hat{l}$ [6]:

$$\hat{l} = \arg \max_{\substack{l \in Z \\ l < 0}} z_{yy}''(x_k, y_{k(c)} + l\lambda). \quad (3)$$

Замечания

Для вычисления $z'(x_k, y_k)$ используем приближенную формулу

$$z_x'(x_k, y_k) \approx \frac{z_{k+1} - z_{k-1}}{x_{k+1} - x_{k-1}}, z_y'(x_k, y_k) \approx \frac{z_{k+1} - z_{k-1}}{y_{k+1} - y_{k-1}} \quad (4)$$

Для вычисления $z''(x_k, y_k)$ используем приближенную формулу

$$z_{xx}''(x_k, y_k) \approx \frac{z_x'(x_{k+1}, y_k) - z_x'(x_{k-1}, y_k)}{x_{k+1} - x_{k-1}} \quad (5)$$

$$z_{yy}''(x_k, y_k) \approx \frac{z_x'(x_k, y_{k+1}) - z_x'(x_k, y_{k-1})}{y_{k+1} - y_{k-1}} \quad (6)$$

Использование методов сплайн-аппроксимации для моделирования биологических объектов

В статье предложено для выделения характерных признаков черт лица использовать модели которые задаются с помощью неравномерных рациональных базисных сплайнов кривых.

Рассмотрим сплайн кривые. Допустим, что задано массив опорных точек $p_0, \dots, p_m$. Необходимо

найти функцию $p(u) = [x(u), y(u), z(u)]^T$,

обозначенную на интервале $u_{\min} \leq u \leq u_{\max}$, такую, что она есть достаточно гладкой и проходит, в определенном смысле, около опорных точек.

Допустим, что есть последовательность узлов $u_0, u_1, \dots, u_n$, такая что:

$$u_{\min} = u_0 \leq u_1 \leq \dots \leq u_n = u_{\max}. \quad (7)$$

Величина $\hat{l}$ в формулах (1), (2) определяется следующим образом [6]:

$$\hat{l} = \arg \min_{L=-1,1} z_{yy}''(x_{k(c)}, y_{k-1(c)} + l\lambda),$$

$$\hat{l} = \arg \max_{l \in Z} z_{yy}''(x_k, y_{k(c)} + l\lambda).$$

При использовании аппроксимации сплайнами функция $p(u)$ имеет вид полинома степени d на интервале между соседними узлами:

$$p(u) = \sum_{j=0}^d c_{jk} u^j, \quad u_k < u < u_{k+1}. \quad (8)$$

Таким образом, чтобы найти сплайн степени d , необходимо будет найти $n(d+1)$ трёхмерный вектор-коэффициентов.

Необходимые для этого уравнения можно получить, рассматривая разного рода ограничения, связанные с непрерывностью функции и критерием близости к опорным точкам. Такой подход к формированию сплайна есть глобальным – необходимо решить систему из $n(d+1)$

уравнение относительно $n(d+1)$ неизвестных, а значит, каждый полученный коэффициент будет зависит от всех опорных точек. Хотя такая методика определения коэффициентов сплайна может обеспечить получение гладкой кривой, которая проходит через заданные опорные точки, но она не очень хорошо согласовывается с спецификой задач компьютерной графики.

Подход выбранный для формирования В-сплайнов состоит из того, чтоб обозначить сплайн в терминах базисных функций, каждая из которых отличная от нуля только на интервале в несколько узлов. Итак, можно записать функцию $p(u)$ в виде:

$$p(u) = \sum_{i=0}^m B_{id}(u) p_i, \quad (9)$$

где каждая функция $B_{id}(u)$ есть полином степени d на интервале в несколько узлов и равняется нулю за границами этого интервала. Существует много способов обозначения базисных функций, но особенное место принадлежит одному из них – методу рекурсивных функций Кокса-де Бура [5]:

$$B_{k,0} = \begin{cases} 1, & \text{если } u_k \leq u \leq u_{k+1} \\ 0 & \text{в другом случае} \end{cases}$$

$$B_{k,d} = \frac{u - u_k}{u_{k+1} - u_k} B_{k,d-1}(u) + \frac{u_{k+1} - u}{u_{k+d+1} - u_{k+1}} B_{k+1,d-1}(u)$$

Выводы

В работе разработан и реализован алгоритм, который позволяет решить задачу моделирования и распознавания мимических проявлений эмоциональных состояний. Анализ полученных результатов дает основание сделать такой вывод: создан алгоритм который реализует

целостную информационную технологию для моделирования и распознавания мимических проявлений эмоционального состояния на лице человека.

Литература

1. Ильин Е.П. Эмоции и чувства/ Ильин Е.П. – СПб: Питер, 2001. – 752 с.
2. Dufour Ph/ Essai sur l'etude le l'Homme considere sous le double point vue de la vie animale et de la vie intellectuelle. 2 vol / Dufour Ph. – Paris: Person, 1883, Режим доступа: <http://www.psychology-online.net/articles>
3. Джемс У. Психология / Джемс У. – М.:Педагогика, 1991 – 368 с.
4. Ланге Г. Душевные движения / Ланге Г. – СП
5. deBoor C.A. Practical Guide to Splines / deBoor C. – New York: Springer-Verlag, 1978. – 392p.
6. Кривонос Ю.Г. Моделирование и анализ мимических проявлений эмоций / Ю.Г. Кривонос, Ю.В. Крак, Г.М. Ефимов // Доповіди НАНУ. – 2008. – №12. – С. 51-55.
7. Миненко А.С. Аналитичность свободной границы в одной задаче осесимметричного течения / А.С. Миненко // Укр. мат. журнал. – 1998. – №12. – С. 1693-1700.
8. Миненко А.С. Проблема минимума одного класса интегральных функционалов с неизвестной областью интегрирования / А.С.Миненко // Мат. физика и нелинейная механика. – 1993. – Вып. 16. – С. 48-52.
9. Миненко А.С. Вариационные задачи со свободной границей / Миненко А.С. – Киев: Наукова думка, 2005. – 354 с.
10. Миненко А.С. Приближенный анализ многомерной конвективной задачи Стефана / А.С. Миненко, А.И. Шевченко // Доповіди НАН України. – 2010. - №4. – С. 30-34.

Міненко А.С. Інформаційні технології при моделюванні та розпізнаванні обличчя людини по його мімічних фотографіям. Робота посвячена розробці методів, моделей, алгоритмів і комп'ютерних засобів, для моделювання і розпізнавання емоційної складової на фотографічних зображеннях обличчя і тривимірних моделях голови людини. Ключові слова: апроксимація, мімічні прояви, емоції, алгоритм, сплайн-апроксимація.

Minenko A.S. Information technology in the modeling and recognition of human face by his facial photos. Work on the development of methods, models, algorithms and computer tools for modeling and recognition of emotional and articulation component the emergence of language, in the event the photographic facial images and three-dimensional models of the human head for playback and recognition of the hard language. Key words: approximation, mimic expression, emotions, algorithm, spline-approximation

Стаття поступила в редакцію 20.11.2016
Рекомендована к публікації д-ром техн. наук В.Н. Павлышом

УДК 004.896

Методика поиска решений в модулях знаний интеллектуальной оболочки

О.В. Морозова

Донецкий национальный технический университет
olmalyavka@gmail.com

Морозова О.В. Механизм срабатывания модулей знаний интеллектуальной оболочки. Рассмотрена проблема проектирования локальной сети со стороны эксперта предметной области (системного администратора). Рассмотрена семантика предметной области. Рассмотрены достоинства и недостатки современных программных продуктов для проектирования локальных сетей. Представлена иерархия модулей знаний. Разработана машина логического вывода срабатывания модулей знаний. Описаны особенности продукционной базы знаний. Выявлены достоинства модульной продукционной базы знаний.

Ключевые слова: локальная сеть, модуль знаний, продукционная база знаний, машина логического вывода.

Введение

Современные информационные потоки, которые представляют собой передачу информации от одного объекта среды к другому, в различных предметных областях многократно возрастают с каждым годом. Поэтому необходимо четкое взаимодействие всех потоков во всех областях.

Для регулирования и упорядочивания информационных потоков служат вычислительные сети различного назначения, от локальных частных сетей, корпоративных до глобальных мировых сетей. Предметной областью является локальная вычислительная сеть, в которой происходит взаимодействие компонентов различных предметных областей.

Локальная вычислительная сеть позволяет объединить персональные компьютеры для получения максимальной эффективности:

- со стороны использованных ресурсов (оборудования, данных, программного обеспечения, процессов);
- для объединения одинаковых функций для выполнения разными узлами сети;
- для соединения всех компьютеров в единое целое без промежуточных устройств.

Главное достоинство сети — это легкость в модификации и расширении готовой локальной вычислительной сети. Для этого служат готовые схемы проектирования.

Современные локальные сети, как промышленного предприятия, так и частного пользования проектируются системным администратором. В данной предметной области (локальные сети) системный администратор

выступает экспертом предметной области и только его опыт и знания гарантируют качественное и эффективное функционирование сегмента сети. Поэтому для облегчения построения и эксплуатации локальной сети необходимо добавление в современные промышленные системы автоматизированного проектирования (САПР) соответствующей интеллектуальной надстройки [1].

Интеллектуальная надстройка объединяет в себе взаимодействие трех видов моделей предметной области, такие как функциональную, процессную и информационную. Логика представления основных структур базы данных — это информационная модель.

Процесс обработки причинно-следственных отношений информации об объекте предметной области является процессной моделью. Функциональная модель организует взаимосвязь объектов предметной области с объектами реального мира.

Цель статьи:

– рассмотреть проблемы проектирования локальных вычислительных сетей со стороны эксперта предметной области и предложить пути их решения с помощью интеллектуальной надстройки;

– рассмотреть алгоритм и метод реализаций организации уровней модулей знаний и блоков представления знаний;

– выявить достоинства и недостатки современных программных продуктов;

– привести структуру взаимодействия блоков;

– рассмотреть семантику представления предметной области и семантику решения задачи проектирования локальной сети.

Выявление достоинств и недостатков современных программных систем для проектирования локальных сетей

Рассмотрим существующие программные продукты Network Notepad, EDraw Network Diagram, LanFlow, 10-Strike LANState. Проанализируем и выявим достоинства и недостатки из которых составим список требований и критериев, которым должна отвечать интеллектуальная оболочка.

Network Notepad программа для создания интерактивных сетевых схем [2].

Достоинства системы Network Notepad:

- быстрое составление физической схемы сети;
- возможность применять значки сетевого оборудования (маршрутизаторы, коммутаторы, и др.);
- поддержка метода Drag&Drop.

Недостатками можно выделить:

- ориентация на сетевых специалистов;
- нет автоматизации проектирования;
- инженер самостоятельно принимает решения при создании проекта локальной сети.

EDraw Network Diagram - это программа для построения диаграмм представление физической схемы локальной сети.

Достоинства системы EDraw Network Diagram:

- основные виды представления локальных сетей;
- топологические схемы сети в виде диаграмм;
- проектирование сетей определенной фирмы производителя Cisco;
- имеет разнообразные примеры для создания диаграмм, символов и шаблонов.

Из недостатков можно выделить:

- ориентация на сетевых специалистов;
- нет автоматизации проектирования;
- инженер самостоятельно принимает решения при создании проекта локальной сети.

10-Strike LANState - программа для администрирования и мониторинга локальных вычислительных сетей.

Достоинства системы 10-Strike LANState:

- наблюдение текущего состояния сети в графическом редакторе;
- мониторинг удаленных станций сети с помощью периодического опроса компьютеров;
- работа с сетью в реальном времени.

Из недостатков можно выделить:

- мало подходит для построения проекта локальной сети;

- большее ориентировано на работу уже с построенной сетью и реальным оборудованием;
- в основном применяется для администрирования существующей сети.

LanFlow - программа для проектирования и документирования сетей.

Достоинства системы LanFlow:

- создание схем локальной сети;
- представление сетевого оборудования в виде 2D и 3D символов;
- добавление клипарта для спецификации сети.

Из недостатков можно выделить:

- для работы с программой необходимы специализированные знания;
- закрытая база компонентов.

Особенности проектирования локальных вычислительных сетей

Задачи, которые необходимо решить эксперту предметной области (системному администратору) [3] для создания оптимальной схемы взаимодействия устройств:

- выбор релевантного оборудования;
- установка связей между всеми устройствами сети;
- назначение прав доступа для каждого устройства сети;
- регулирование передачи пакетов в сети между устройствами;
- назначение адресов узлов в сети, проверка на уникальность адресов в данном сегменте сети;
- отслеживание ошибок настройки сети и работоспособности всех компонентов без сбоев;
- рассчитать возможности перехода на другое телекоммуникационное оборудование;
- обеспечение защиты данных в сети.

Для проектирования физической модели локальной сети (см. рис. 1) системному администратору придется учитывать много параметров:

- количество этажей;
- количество комнат на этаже с параметрами комнат (длина, ширина и высота этажа);
- количество рабочих мест;
- сетевое оборудование для соединения устройств сети;
- серверное оборудование;
- и др.

Такой объем выполняемой работы проектирования (или модернизации) одной сети будет выполняться системным администратором в течении нескольких дней и в несколько этапов. На первом этапе будет определена топология сети и разработан порядок расстановки рабочих станций. На втором этапе осуществлён подсчет параметров комнат и этажей для оборудования, и

выбор оптимального сетевого оборудования (коммутаторов, маршрутизаторов, серверного оборудования и т.д.). На третьем этапе будет

рассчитана экономическая составляющая проекта. Заключительный этап - это монтаж сети.

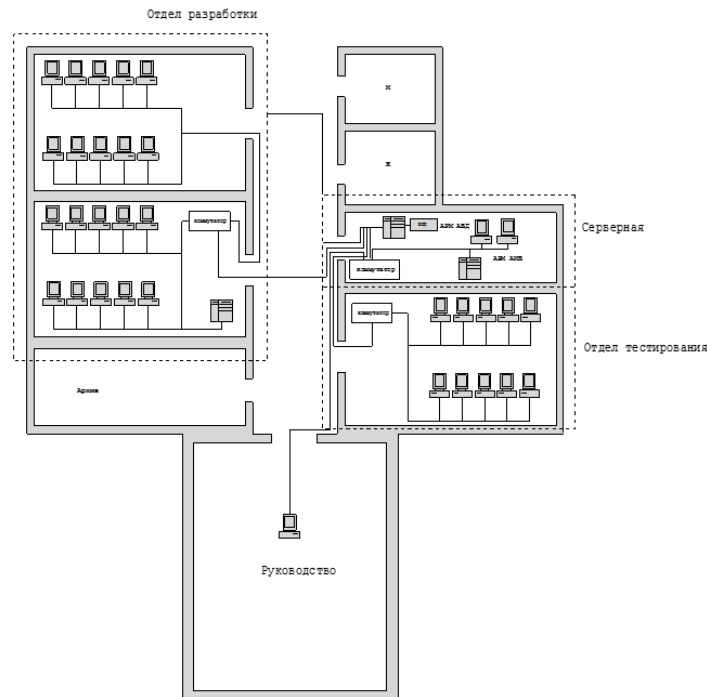


Рисунок 1 – Модели локальной сети

Для ускорения и эффективного проектирования сети можно применить САПР с интеллектуальной надстройкой именно на первом и втором этапах. Так как существует множество САПР различного назначения от графического представления до просчета экономичности [1], рассмотрим формирование и срабатывание базы знаний из готовых прототипов сети и модернизации их до конкретных сетей общего и частного пользования.

Организация продукционной базы знаний

Оптимизация построения локальной сети в интеллектуальной оболочке происходит так же в два этапа. Первый этап - организация диалога в виде набора правил и предъявления требований к структуре сети (самоинтервью) [4]. На втором этапе осуществляется автоматическая параметризация и выбор оптимального оборудования.

Такой метод проектирования сетей позволяет абстрагироваться от конкретного оборудования и позволяет построить структурную модель сети.

Для выявления всей полноты информации о предметной области и описания ее в иерархическом структурированном виде будем использовать модульный принцип организации базы знаний. При таком подходе

происходит покрытие всех требований, накладываемых на предметную область.

Дескриптивная модель позволяет описать топологию взаимодействия сети и информацию, передаваемую в данной локальной сети. Она позволяет выбирать оптимальный метод решения поставленной задачи и построить формальное представление задачи.

Формирование базы знаний будем производить на основании самоинтервью. Множество фактов хранимых в базе знаний имеют тип и прототип [5]. Прототип – это отдельно взятое решение, которое имеет вид факта. Факт может быть единичный, базовый, или – сложный. Сложный факт задает множество фактов, если решение формируется как набор из базового множества фактов или модификация этого множества. Множество решений, составляющих тип - это И-ИЛИ-дерево. В типе есть неизменная часть, т.е. - системообразующая, которая принадлежит любому прототипу, и – факультативная часть, как набор разнообразных частей.

Тип по отношению к типу блока – это множество:

- структур блока, если блок не базовый;
- функций, если блок базовый.

Используется модульная база знаний [6].

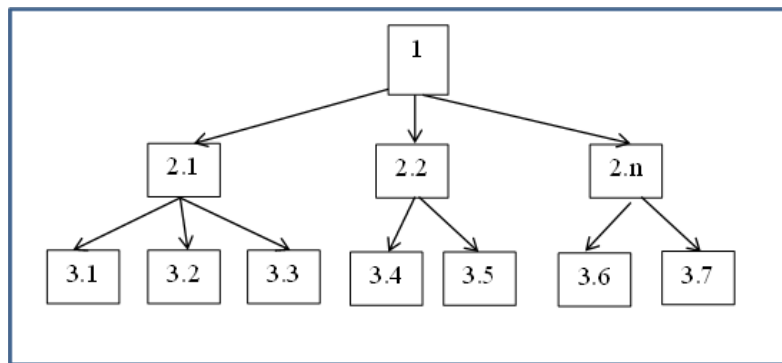


Рисунок 2 – Структура модулей знаний

Тип – это модуль знаний, прототип – отдельно взятое решение.

Если модуль знаний делится на подмодули (внешняя граница, внутренняя и т.д.), то так же имеется и набор фактов - база данных этого же уровня. Совокупность прототипов, составляющих тип, может явно не задаваться перечислением и не входить в базу данных в состав модуля знаний, например, типа блока. Тогда в глобальной базе знаний имеется только тип блока как модель знаний.

Список прототипов или фактов всех уровней сложности, включая готовые блоки в сборе, варианты внешних границ, варианты внутренних наборов подблоков, варианты групп связей разных форм (внутренние и т.д.) формируется в процессе вывода.

Каждый модуль знаний представляет собой независимую структурную единицу. При этом дробление каждого модуля возможно на подмодули. Связь устанавливается в зависимости от границ входов и выходов. Если в модуле отсутствуют выходы, он является конечным. Для хранения всех границ используется таблица границ входов и выходов (см. рис. 3). Для обозначения связей между модулями используется таблица связей. В ней указываются порядковые номера входов и выходов.

Система поддерживает два типа связей:
- звезда;
- последовательное соединение.

Представление метода организации логической машины вывода

При срабатывании интеллектуальной надстройки, генерируется базовый блок первого уровня. В зависимости от условий, он имеет определённое количество входов и выходов [7].

В течении работы логической машины вывода, внутри базового блока синтезируются подблоки. Так же происходит соединение

входов и выходов внутренней границы базового подблока первого уровня, с внешними границами синтезированных подблоков второго уровня.

На следующем этапе работы логической машины происходит переход по связи из блока первого уровня в блок второго уровня. Теперь мы можем произвести синтез подблоков третьего уровня внутри блока второго уровня.

Далее происходит соединение входов и выходов внутренней границы блока второго уровня с внешними границами синтезированных блоков третьего уровня. Данная операция синтеза может повторяться, до тех пор, пока не будет достигнут конечный блок, не имеющий внутренних границ входов и выходов (см. рис. 3).

При создании базы знаний, пользователь сам указывает тип связи. После подачи команды, задаётся узловой блок, с которым устанавливаются все последующие связи. При поступлении нового блока, указатель смещается на него, и пока не будет дана команда отмены, данный блок является узловым (см. рис 4 и рис 5).

В таблице связей указываются переходы от одного блока к другому. Для учёта берутся порядковые номера записей из таблицы входов и выходов, и заносятся в поля. Первое поле указывает на источник связи, а второе на приёмник.

Использование модульной базы знаний и минимального набора операторов позволяют редактировать непосредственно каждый блок. При вводе новых продукций необходимо придерживаться точного порядка следования правил. При работе машины вывода, происходит построчное чтение базы. Если допустить ошибку при вводе правил, может не сработать условие и либо произойдёт аварийное завершение работы машины вывода, либо в

построенной топологии будут присутствовать ошибки.

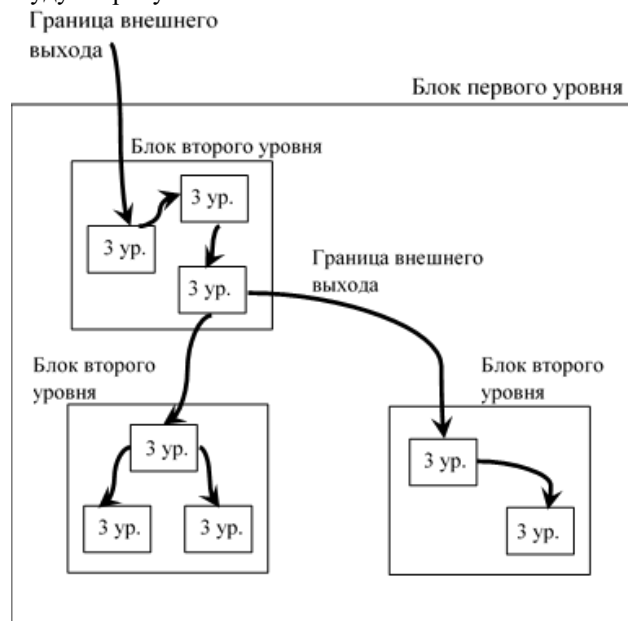


Рисунок 3 – Схема взаимодействия уровней связи блоков

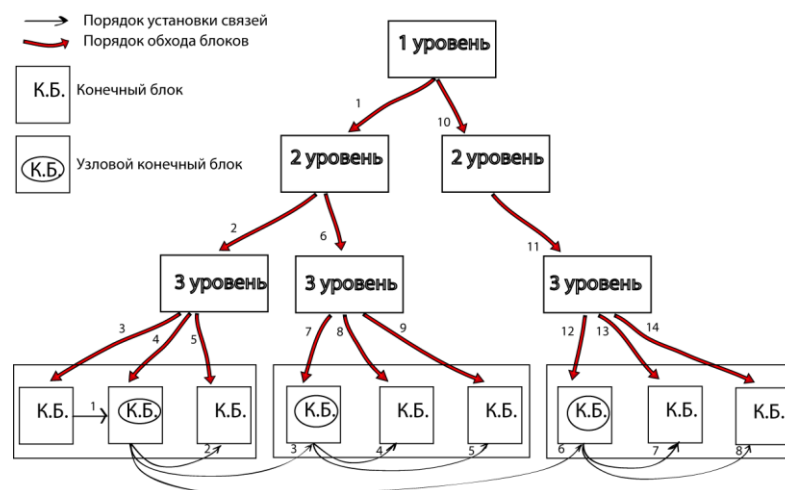


Рисунок 4 – Модель связи блоков типа звезда

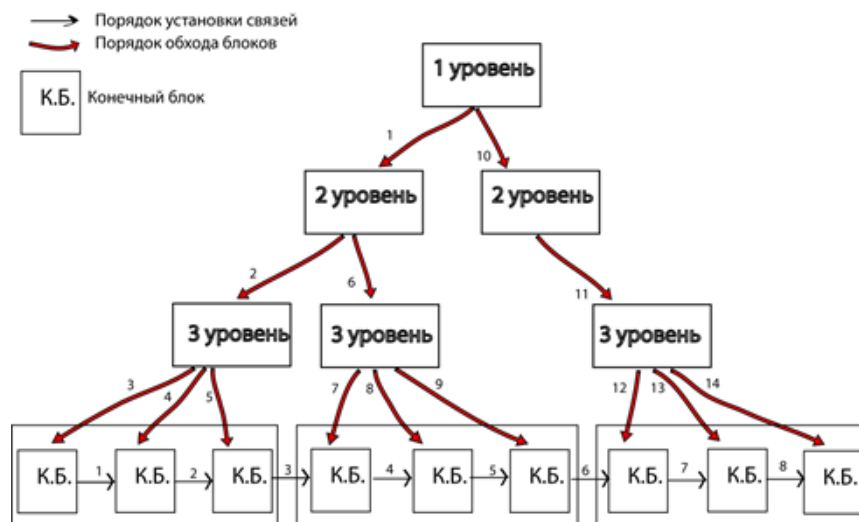


Рисунок 5 – Модель связи блоков с последовательными связями

При работе машины вывода, происходит построчное чтение продукции в текущем модуле знаний. В зависимости от типа первого символа, выполняются определённые продукции. При переходе в новый подблок, работа с текущей базой останавливается. Возобновление работы происходит после возврата в блок, когда обработаны все нижестоящие подблоки. Работа с блоком возобновляется со следующей строки после команды перехода на блок нижнего уровня.

Во время работы с блоками, возможен доступ к структурам машины вывода. В данных структурах хранится информация о доступных блоках, количестве загрузок каждого блока, в каком блоке находимся в текущий момент.

При возникновении ситуации, когда модуль полностью прочитан, но остались необработанные продукции, работа машины вывода останавливается, и пользователю выдаётся ошибка о нарушении целостности базы.

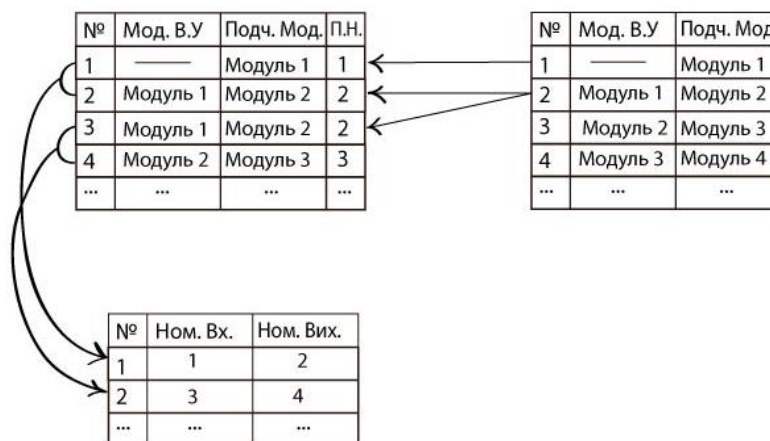


Рисунок 5 – База данных взаимодействия модулей знаний

Выводы

В статье был проведен анализ проблемы проектирования локальных вычислительных сетей на основе эксперта предметной области. Проанализированы современные программные системы для проектирования сети и выявлены в них достоинства и недостатки. Предложен процесс упрощения и эффективного проектирования локальных сетей при

использовании интеллектуальной надстройки для систем автоматизированного проектирования. Была описана структура продукционной базы знаний в виде иерархии модулей знаний. Представлен механизм логической машины вывода и организация взаимосвязей блоков данных в интеллектуальной надстройке. Выявлены достоинства модульной продукционной базы знаний:

- отдельный модуль знаний, позволяет организовать быстрый поиск в иерархии модулей знаний и по дереву решений;

- возможность многократного использования модуля знаний на разных уровнях логической машины вывода.

Перспективой дальнейшего развития работы является создание и описание нового метода проектирования локальных сетей на основе топология «звезда» и «иерархическая звезда». Так же необходимо будет учесть функциональные зависимости объекта сети с ее структурными элементами.

Литература

1. Крылов К.С. Синтез проектов локальных вычислительных сетей с использованием экспертных методик/ К.С. Крылов, А.В. Григорьев // Труды конференции ИУС и КМ -2011 том третий. – Донецк, 2011. – 200 с.

2. Автоматизированное проектирование СКС и ЛВС NetWizard [Электронный ресурс]. - Электрон. дан. – Режим доступа <http://netwizard.ru/>

3. Олифер, Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 3-е издание / В. Г. Олифер // СПб.: Питер, 2006 – С. 45-79.

4. Григорьев, А.В. Создание интеллектуальной оболочки по методикам проектирования локальных сетей при использовании проектировщика как эксперта по знаниям / А.В. Григорьев, О.В. Морозова // Наукові праці ДонНТУ, Серія: «Проблеми моделювання та автоматизації проектування». Вип. №1(10)-2(11). – Донецьк, 2012. – С. 144-155.

5. Солодовников В.В. Теория сложности и проектирование систем управления/ В.В. Солодовников, В.И. Тумаркин// М.: Наука, 1990. – 168 с.

6. Ильянсов Б. Г. Модульная структура базы знаний экспертной системы «Выбор специальности при поступлении в вуз»/ Б. Г. Ильясов, Е. Б. Старцева, Н. Р. Янгуразова/ Электронный журнал Нефтегазовое дело, 2006 – С. 4-5.

7. Морозова О.В. Анализ методов построения экспертных систем в продукционных инструментальных оболочках / О.В. Морозова, А.В. Григорьев // Наукові праці ДонНТУ. – Донецьк, 2012. – Вип. 16(204). – С. 180-191.

8. Григорьев А.В. Анализ существующих способов создания интерфейса «языки формальных спецификаций — проблемно-ориентированные языки» / Григорьев А.В., Морозова О.В. // Сборник научных трудов донецкого национального технического университета. – Донецк, 2011. – № 14. – С. 270 – 275.

9. Григорьев А.В. Построение двухсторонних трансляторов в задаче создания интеллектуальных надстроек над проблемно-ориентированными САПР / Григорьев А.В., Морозова О.В. // Сборник трудов XI международной научной конференции им. Т.А. Таран. - Киев: Просвита, 2011. - С. 68-75.

10. Люгер, Д. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем. 4-е издание / Пер. с англ. – М.: «Вильямс», 2003 -С. 219-262.

11. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский СПб.: Питер, 2000 –С. 71-73.

12. Голдовский, П. Knowledgeware: как задействовать самые ценные корпоративные активы [Электронный ресурс] / Павел Голдовский, Андрей Бубнов // САПР и графика. - Электрон. журн. - №4'2001. - Режим доступа к журн.:

<http://www.sapr.ru/article.aspx?id=7125&iid=291>

Морозова О.В. Механизм срабатывания модулей знаний интеллектуальной оболочки. Рассмотрена проблема проектирования локальной сети со стороны эксперта предметной области (системного администратора). Рассмотрена семантика предметной области. Рассмотрены достоинства и недостатки современных программных продуктов для проектирования локальных сетей. Представлена иерархия модулей знаний. Разработана машина логического вывода срабатывания модулей знаний. Описаны особенности продукционной базы знаний. Выявлены достоинства модульной продукционной базы знаний.

Ключевые слова: локальная сеть, модуль знаний, продукционная база знаний, машина логического вывода.

Morozova OV Response Mechanism of intellectual shell of knowledge modules. The problem of designing a LAN by subject area (system administrator) expert. We consider the semantics of the subject area. The advantages and disadvantages of modern software products for the design of local networks. It

presents a hierarchy of knowledge modules. A car trip inference of knowledge modules. The features production base of knowledge. Revealed the dignity of modular production base of knowledge.

Keywords: *LAN, knowledge module produktsionnaja knowledge base, inference engine.*

*Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко*

УДК 621.376.57

Исследование использования дельта-преобразований для определения местоположения беспилотного летательного аппарата

Л.В. Пирская, П.П. Кравченко
Южный федеральный университет
lpirskaya@sfnedu.ru, kravchenkopp@sfnedu.ru

Пирская Л.В., Кравченко П.П. Исследование использования дельта-преобразований для определения местоположения беспилотного летательного аппарата. В работе рассматриваются возможности решения задачи определения местоположения беспилотного летательного аппарата (БПЛА) специализированным вычислительным устройством. В постановке задачи рассматривается определение координат БПЛА на основе координат расположенных на относительно близком расстоянии от начала локальной системы координат групп четверок маяков и дальности от БПЛА до маяков. На основании имеющихся данных для каждой группы формируется в стандартном для дальномерной навигации система четырех уравнений, которая преобразовывается к СЛАУ третьего порядка. Решение данной СЛАУ с непрерывными переменными свободными членами определяет координаты БПЛА. В работе рассмотрены особенности алгоритмизации решения задачи определения местоположения БПЛА при наземной подготовке предложенного задания и решении задачи на борту БПЛА в режиме реального времени. Проведено исследование возможности использования дельта-преобразований второго порядка для решения СЛАУ в поставленной задаче на различных примерах размещения маяков. Полученные результаты экспериментов с использованием компьютерного моделирования подтверждают возможность получения решения СЛАУ за одну итерацию в установившемся процессе с временным шагом, представляющим практический интерес, а начало установившегося процесса обеспечивается на достаточно большой удаленности от начала координат.

Ключевые слова: Определение координат БПЛА, решение СЛАУ, дельта-преобразование второго порядка, бортовое специализированное вычислительное устройство.

Введение

В настоящее время задача проектирования бортовых высокоточных интегрированных систем управления и навигации беспилотными летательными аппаратами (БПЛА) является одной из современных перспективных задач научно-исследовательского и инженерно-технического характера. Решение бортовой задачи локальной навигации, в частности, определение координат БПЛА, осуществляется в режиме реального времени в условиях одновременного выполнения других бортовых задач при определенном временном шаге дискретизации процессов управления [1,2]. В связи с этим весьма актуальной является проблема минимизации временных и аппаратных затрат для решения поставленной задачи в пределах временного шага, а также обеспечение минимальной задержки решения относительно начала временного шага, что связано с ограничением влияния задержки на ошибку управления. Для решения задач с ограничениями отмеченного характера возможности

высокопроизводительных вычислителей общего назначения ограничены.

Таким образом, возникает необходимость в создании специализированного вычислительного устройства, которое должно обеспечивать высокопроизводительное и экономичное по аппаратным затратам решение задачи локальной навигации, и в частности, определение местоположения БПЛА в реальном масштабе времени. Для решения данной задачи возможно использовать во многих случаях для вычисления координат БПЛА не только итерационные методы (например, метод наименьших квадратов, который является стандартом в приемниках спутниковых навигационных систем), но и конечных алгебраических методов, основанных на приведении системы навигационных уравнений к системе линейных алгебраических уравнений и имеющих существенно более низкую трудоемкость реализации. В рамках данной работы рассматривается возможность построения специализированного вычислительного устройства на основе использования дельта-

преобразований, позволяющих при использовании итерационного метода решения СЛАУ организовывать вычислительный процесс с исключением операций многоразрядного умножения и с получением результата за одну итерацию установившегося процесса [3-8]. Предварительные исследования [7-8] показали преимущества данного подхода по длительности выполнения одной итерации и итерационного процесса в целом по сравнению с реализацией известного метода «простой итерации» [9-12] примерно в 1,7 раза. При проектировании данного специализированного устройства с использованием ПЛИС особенности алгоритмического обеспечения позволяют достигать наиболее высоких качественных показателей производительности при минимизированных затратах ресурсов (оборудования) по сравнению с известными [13-14].

С точки зрения эффективного использования ресурсов бортовых вычислительных средств представляет интерес такая организация вычислительного процесса, когда обработка информации на уровне одной итерации выполняется с достаточной точностью, высоким быстродействием и с предельно большим временным шагом. В таких условиях возможно предъявление наиболее низких требований по частоте формирования данных, характеризующих изменяющиеся свободные члены СЛАУ (например, частоте формирования расстояний от БПЛА до маяков локальной системы навигации [1,2]), а также к производительности вычислительных средств с учетом возможности одновременной реализации других алгоритмов и программ. Актуальность эффективного решения СЛАУ даже невысоких порядков в рассматриваемых условиях резко возрастает, когда необходимо одновременно решать большое количество СЛАУ, что имеет место при использовании маяковой системы локальной навигации с одновременным решением задачи обеспечения целостности [15].

Таким образом, целью работы является исследование вопросов проектирования высокопроизводительных экономичных по аппаратным ресурсам специализированных вычислителей для одновременного решения большого количества СЛАУ на основе дельта-преобразований второго порядка, соответствующих задаче определения местоположения БПЛА с использованием четырех маяков.

Постановка задачи определения местоположения БПЛА

Одна из задач определения местоположения БПЛА базируется на

использовании нескольких разнесенных в пространстве маяков, расположенных на относительно близком расстоянии от начала локальной системы координат. Бортовой вычислитель получает значения дальностей от БПЛА до маяков. На основании имеющихся координат маяков и полученных дальностей можно сформировать навигационные определения в стандартном для дальномерных навигационных систем виде [16]:

$$(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = D_i^2, \quad (1)$$

где x, y, z – координаты БПЛА; (x_i, y_i, z_i) – координаты i -го маяка в выбранной системе координат; D_i – дальность до i -го маяка. Решение системы уравнений для всех маяков позволяет определить местоположения БПЛА (x, y, z) в пространстве.

Решение данной задачи на борту БПЛА включает 2 этапа.

Первым этапом является определение текущих координат БПЛА путем решения систем уравнений, каждое из которых базируется на данных размещения соответствующих этой системе четырех маяков и дальностей БПЛА до этих маяков.

Вторым этапом является решение задачи целостности, обеспечивающей достоверность получаемых результатов с учетом возможных нарушений в работе маяков. Исходными данными для задачи целостности являются формируемые множественные значения координаты БПЛА, полученные на первом этапе. Для успешного решения данной задачи необходимо использование большого количества групп маяков, позволяющих независимо формировать по каждой группе координаты БПЛА. В работе, собственно, решение задачи целостности не рассматривалось.

Определение местоположения БПЛА с использованием четырех маяков

В постановке задачи рассматриваются:

- координаты четырех маяков в декартовой системе координат: (X_1, Y_1, Z_1) , (X_2, Y_2, Z_2) , (X_3, Y_3, Z_3) , (X_4, Y_4, Z_4) ; данные координаты формируются при установке маяков путем выполнения измерений на местности;

- расстояния от каждого маяка до БПЛА: D_1, D_2, D_3, D_4 ; данные координаты формируются в полете с использованием специальной бортовой аппаратуры.

Необходимо определить координаты X_p, Y_p, Z_p БПЛА.

В соответствии с (1) квадраты расстояний от 4-х маяков до БПЛА имеют вид:

$$\begin{cases} (X_p - X_1)^2 + (Y_p - Y_1)^2 + (Z_p - Z_1)^2 = D_1^2 \\ (X_p - X_2)^2 + (Y_p - Y_2)^2 + (Z_p - Z_2)^2 = D_2^2 \\ (X_p - X_3)^2 + (Y_p - Y_3)^2 + (Z_p - Z_3)^2 = D_3^2 \\ (X_p - X_4)^2 + (Y_p - Y_4)^2 + (Z_p - Z_4)^2 = D_4^2 \end{cases}$$

Раскрываем скобки и получаем:

$$\begin{cases} X_p^2 - 2X_pX_1 + X_1^2 + Y_p^2 - 2Y_pY_1 + Y_1^2 + Z_p^2 - 2Z_pZ_1 + Z_1^2 - \\ X_p^2 - 2X_pX_2 + X_2^2 + Y_p^2 - 2Y_pY_2 + Y_2^2 + Z_p^2 - 2Z_pZ_2 + Z_2^2 - \\ X_p^2 - 2X_pX_3 + X_3^2 + Y_p^2 - 2Y_pY_3 + Y_3^2 + Z_p^2 - 2Z_pZ_3 + Z_3^2 - \\ X_p^2 - 2X_pX_4 + X_4^2 + Y_p^2 - 2Y_pY_4 + Y_4^2 + Z_p^2 - 2Z_pZ_4 + Z_4^2 = \end{cases}$$

В рамках данной работы с целью создания более благоприятных условий для формирования нормы достаточной ограниченности переход к СЛАУ третьего порядка предлагается осуществлять путем следующей последовательности вычитаний: (2.1)-(2.2), (2.2)-(2.3), (2.3)-(2.4). В результате получаем:

$$\begin{cases} 2X_p(X_2 - X_1) + 2Y_p(Y_2 - Y_1) + 2Z_p(Z_2 - Z_1) = \\ 2X_p(X_3 - X_2) + 2Y_p(Y_3 - Y_2) + 2Z_p(Z_3 - Z_2) = \\ 2X_p(X_4 - X_3) + 2Y_p(Y_4 - Y_3) + 2Z_p(Z_4 - Z_3) = \\ \begin{cases} = D_1^2 - D_2^2 + X_2^2 + Y_2^2 + Z_2^2 - X_1^2 - Y_1^2 - Z_1^2 \\ = D_2^2 - D_3^2 + X_3^2 + Y_3^2 + Z_3^2 - X_2^2 - Y_2^2 - Z_2^2 \\ = D_3^2 - D_4^2 + X_4^2 + Y_4^2 + Z_4^2 - X_3^2 - Y_3^2 - Z_3^2 \end{cases} \end{cases} \quad (3)$$

Сходимость итерационного решения СЛАУ обеспечивается в соответствии с условиями для метода простой итерации [9-12], которые формируются на основе данных о размещении маяков.

Преобразуем (3) к виду:

$$\begin{cases} (X_2 - X_1)X_p + (Y_2 - Y_1)Y_p + (Z_2 - Z_1)Z_p = \\ (X_3 - X_2)X_p + (Y_3 - Y_2)Y_p + (Z_3 - Z_2)Z_p = \\ (X_4 - X_3)X_p + (Y_4 - Y_3)Y_p + (Z_4 - Z_3)Z_p = \\ \begin{cases} = \frac{1}{2}(D_1^2 - D_2^2 + X_2^2 + Y_2^2 + Z_2^2 - X_1^2 - Y_1^2 - Z_1^2) \\ = \frac{1}{2}(D_2^2 - D_3^2 + X_3^2 + Y_3^2 + Z_3^2 - X_2^2 - Y_2^2 - Z_2^2) \\ = \frac{1}{2}(D_3^2 - D_4^2 + X_4^2 + Y_4^2 + Z_4^2 - X_3^2 - Y_3^2 - Z_3^2) \end{cases} \end{cases} \quad (4)$$

Далее для определения координат (X_p, Y_p, Z_p) БПЛА необходимо решать полученную систему уравнений (4) относительно X_p, Y_p, Z_p .

Особенности решения задачи определения местоположения БПЛА

В решении задачи определения местоположения БПЛА имеет место два этапа:

наземная подготовка предложенного задания: при этом выполняются все математические преобразования, которые могут минимизировать алгоритм по вычислительной трудоемкости при решении задачи на борту БПЛА, а также определяются те группы маяков по четыре, которые соответствуют условиям обеспечения сходимости [9-12];

решение на борту подготовленной на земле задачи в обстановке реального времени.

В условиях наземной подготовки целесообразно устанавливать $n \gg 4$ маяков, что как было отмечено в разделе 1, связано в первую очередь с обеспечением эффективного решения задачи целостности.

Число сочетаний из n маяков по m вычисляется по формуле:

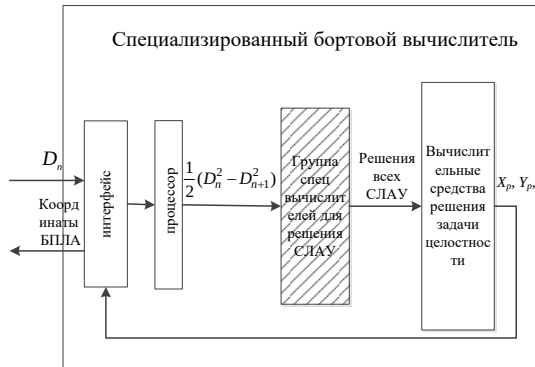
$$C_n^m = \frac{n!}{m!(n-m)!}$$

Предположим, что установлено $n=8$ маяков, тогда число сочетаний данных маяков применительно к данному алгоритму будет составлять $C_8^4 = 70$, но использование всех комбинации не обязательно.

В рамках данной работы рассматриваются возможности одновременного (параллельного) решения множества СЛАУ для 4-х маяков на основе специализированного вычислителя с алгоритмизацией решения на базе дельта-преобразований второго порядка [7-8].

На рисунке 1 представлена укрупненная структурная схема специализированного бортового вычислителя, отражающая особенности возможной реализации решения задачи определения местоположения БПЛА, где в виде заштрихованного блока представлена группа специализированных вычислителей для решения СЛАУ на основе дельта-преобразований второго порядка и переменного кванта, D_n - расстояния от каждого n -ого маяка до БПЛА.

В бортовой вычислитель при наземной подготовке загружаются разности координат маяков и значения известных сумм квадратов свободных членов системы (4). В полете в режиме реального времени на универсальный бортовой вычислитель (процессор на рисунке 1) поступают значения дальностей $D_1^2, D_2^2, D_3^2, D_4^2$, и далее в представленном на рисунке 1 заштрихованном блоке осуществляется решение СЛАУ на основе алгоритма дельта-преобразований второго порядка [7-8].



3

4 Рисунок 1 - Увеличенная структурная схема специализированного бортового вычислителя.

Алгоритм решения СЛАУ с использованием дельта-преобразований второго порядка и переменного кванта

Рассмотрим решение СЛАУ, содержащую матрицу постоянных коэффициентов и в общем случае переменные свободные члены, удовлетворяющую условиям сходимости для метода простой итерации [9-12], и имеющую вид:

$$BY^*(t) = G(t). \quad (5)$$

Преобразуем систему:

$$Y^*(t) = AY^*(t) + D(t).$$

Переходим к форме записи с введением невязки $z(t)$ и использованием итерационного метода:

$$z(t) = Y(t) - AY(t) - D(t). \quad (6)$$

В приведенных системах $B = [b_{ij}]$, $A = [b_{ij} / b_{rr}]$ - матрицы коэффициентов размерности $n \times n$; $G(t)$, $D(t)$ - вектор-столбцы свободных членов системы (в частном случае для системы с постоянными свободными членами $G(t) = G = [g_r]$, $D(t) = D = [g_r / b_{rr}]$); $Y^*(t)$ - вектор-столбец неизвестных системы; $z(t)$, $Y(t)$ - вектор-столбцы невязок и приближенных значений неизвестных; t - независимая переменная; $\det A \neq 0$.

Алгоритм (7) параллельного решения СЛАУ (5) с использованием дельта-преобразования второго порядка и переменного кванта представим в следующей разностной форме для i -го шага при начальных условиях $Y_{r01} = 0$, $\nabla Y_{r01} = 0$, $z_{r01} = -D_{r01}$, $\nabla z_{r01} = -\nabla D_{r01}$, $r = \overline{1, n}$ [3, 7-8]:

- демодуляция:

$$\nabla^2 Y_{ril} = c_l^* \Delta_{ril}; \quad (7.1)$$

$$\nabla Y_{ril} = \nabla Y_{r(i-1)l} + \nabla^2 Y_{ril}; \quad (7.2)$$

$$Y_{ril} = Y_{r(i-1)l} + \nabla Y_{ril}; \quad (7.3)$$

$$r = \overline{1, n}, i = 1, 2, \dots, R_l, l = 1, 2, \dots, P;$$

- формирование второй разности преобразуемой переменной:

$$\nabla^2 y_{ril} = \sum_{j=1}^n a_{rj} c_l^* \Delta_{jil} + \nabla^2 D_{ril}; \quad (7.4)$$

- формирование значений невязок:

$$\nabla^2 z_{ril} = \nabla^2 Y_{ril} - \nabla^2 y_{ril}; \quad (7.5)$$

$$\nabla z_{ril} = \nabla z_{r(i-1)l} + \nabla^2 z_{ril}; \quad (7.6)$$

$$z_{ril} = z_{r(i-1)l} + \nabla z_{ril}; \quad (7.7)$$

- формирование значений переключающих функций и знаков квантов вторых разностей:

$$F_{ril} = z_{ril} + 1.5 \nabla z_{ril} + (0.5 \nabla z_{ril}^2 / c_l - 0.125 c_l) \text{sign}(\nabla z_{ril}); \quad (7.8)$$

$$\Delta_{r(i+1)l} = -\text{sign} F_{ril}; \Delta_{ril} \in \{+1, -1\}, \quad (7.9)$$

В алгоритме (7) c_l^* - вес модуля кванта преобразования на l -ом итерационном цикле ($c_l^* > 0$), P - количество циклов, выполняемых при постоянных по модулю значениях квантов, R_l - количество итераций в цикле. Кроме того, для стыков участков соседних циклов при решении СЛАУ с переменными свободными членами используются соотношения: $Y_{r0l} = Y_{rR(l-1)}$;

$$z_{r0l} = z_{rR(l-1)}.$$

Сущность процесса решения системы (5) на основе алгоритма (7) состоит в том, что задаются начальные условия Y_{r0} , ∇Y_{r0} , D_{r0} , ∇D_{r0} , $r = \overline{1, n}$ (в частном случае, например, $Y_{r0} = 0$, $\nabla Y_{r0} = 0$, тогда, соответственно, $z_{r0} = -D_{r0}$, $\nabla z_{r0} = -\nabla D_{r0}$, $r = \overline{1, n}$) и организуется итерационный (переходный) процесс решения до вхождения в установившийся процесс, когда $|z_{riP}| \leq z_{steady}$, $z_{steady} > 0$, $r = \overline{1, n}$, где z_{steady} - достаточно малые, соответствующие обеспечению заданной точности решения системы величины. Дискретные значения D_{ri} , $r = \overline{1, n}$, $i = 1, 2, \dots$ в алгоритме (7) предполагаются численно определенными на каждом шаге [3].

В работах [7-8] разработаны целочисленные оценки параметров алгоритма (7), определяющие способ задания последовательности значений переменных квантов в циклах. Предложены следующие соотношения для квантов соседних циклов (s - степень минимального кванта, $s \in N$):

$$\text{для } R_{\text{int},1} = 4 \quad c_{P_{\text{int},1}-l}^* = 2^{2(l-1)-s}, \quad l = \overline{P_{\text{int},1}, 1};$$

$$\text{для } R_{\text{int},2} = 8 \quad c_{P_{\text{int},2}-l}^* = 2^{3(l-1)-s}, \quad l = \overline{P_{\text{int},2}, 1}$$

В работах [7-8] введены и теоретически обоснованы эффективные условия окончания итерационного процесса в текущем цикле алгоритма (7):

$$\begin{aligned} \operatorname{sign}\left(\frac{z_{r(i+1)l}}{c_l^*}\right) &= -\operatorname{sign}\left(\frac{z_{ril}}{c_l^*}\right), \\ \operatorname{sign}\left(\frac{|z_{ril}|}{c_l^*}\right) &= -\operatorname{sign}\left(\frac{|z_{ril}|}{c_l^*} - \operatorname{sign}\left(\frac{|z_{ril}|}{c_l^*}\right) \cdot \frac{|z_{ril}|^*}{c_l^*}\right); \\ r &= \overline{1, n}, \quad i = 1, 2, \dots, R_l; \quad l = 1, 2, \dots, P. \end{aligned}$$

Исследование использования дельта-преобразований второго порядка для решения СЛАУ в задаче определения местоположения БПЛА

Проводилось исследование использования дельта-преобразований второго порядка для решения СЛАУ в задаче определения местоположения БПЛА на различных примерах размещения маяков. Особое внимание уделялось высоте расположения маяков, так как с уменьшением этих высот в большей мере проявляется влияние возмущений на ошибку итерационного процесса на каждом шаге.

Кроме того, исследование проводилось с учетом двух возможных способов организации вычислительного процесса:

1. Значения текущих дальностей поступают на каждом текущем шаге как переходного, так и установившегося процессов; длительность всех шагов (итераций) фиксирована.

2. Первый временной шаг решения используется для реализации переходного процесса при фиксированных, соответствующих этому шагу времени значениях дальностей (данный режим реализуется при наличии возможности передачи для данной задачи необходимых вычислительных ресурсов на шаге). После завершения данного переходного процесса продолжается решение на последующих временных шагах с реализацией одной итерации на каждом из этих шагов. Предполагается автоматическое выявление момента завершения переходного процесса и формирование соответствующего признака в решении задачи управления БПЛА по установлению наименьшего значения кванта преобразования.

Следует обратить внимание на то, что в основе организации вычислительного процесса по второму способу реализуются переходные процессы, которые по критерию сходимости соответствуют рассмотренным в [9-12] положениям. В случае использования данного способа организации переходного процесса при работе с переменными свободными членами представляются возможности сокращения количества временных шагов решения (но не

количества итераций).

Исследование использования дельта-преобразований второго порядка для решения СЛАУ в задаче определения местоположения БПЛА будем рассматривать на отдельных группах маяков по четыре на двух типах задач. Первая задача – определение координат при большой начальной высоте БПЛА и большой скорости, вторая задача – при посадке БПЛА.

В первом типе задач рассмотрены три варианта расположения 4-х маяков:

(-200, -400, 20), (450, -100, 120), (120, 400, 170), (-100, 450, 420); (8)

(-200, -400, 5), (450, -100, 25), (120, 400, 0), (-100, 450, 25); (9)

(100, -40, 1), (150, -100, 0), (10, 400, 0), (-15, 450, 10). (10)

Вариант (8) отличается от варианта (9) значительным снижением высоты расположения маяков. Вариант (10) имеет отличный вариант расположения маяков в плоскости ХОУ, а также имеет предельно низкие высоты маяков.

Начальная высота БПЛА составляет $H_0 = 17000$ м., скорость $V_1 = 700$ м/с и $V_2 = 300$, угол к поверхности земли 60° .

Решение всех СЛАУ, соответствующих представленным выше координатам маяков и составленным по системе (4), осуществлялось при обеспечении одинаковой точности 2^{-14} , характеризующейся значением 0,25 м.

В таблицах 1, 2 представлены результаты, характеризующие временные шаги ∇t_2 и высоту H_2 в метрах при использовании алгоритма на базе дельта-преобразований второго порядка [7-8]. Параметр H_2 отражает высоту, начиная с которой обеспечивается решение СЛАУ за одну итерацию в установившемся процессе с заданной одинаковой максимальной точностью решения для трех вариантов расположения маяков (8), (9), (10). В таблице 1 организация вычислительного процесса осуществлялась на основе первого способа, в таблице 2 – на основе второго способа. В таблицах 1, 2 в скобках обозначены номера описаний координат расположения маяков (8), (9), (10).

Таблица 1. Результаты экспериментов решения СЛАУ задачи определения местоположения БПЛА на основе первого способа организации вычислительного процесса.

∇t_2	Высота БПЛА для дельта-преобразований второго порядка, H_2					
	$V_1 = 700$ м/с			$V_1 = 300$ м/с		
	(8)	(9)	(10)	(8)	(9)	(10)
0,00	1635	1587	1550	1684	1627	1651
5	7	7	7	5	7	0
0,01	1570	1355	1476	1660	1601	1562
	7	9	4	0	0	8

0,02 5	1436 3	1101 8	9652	1539 7	1384 7	1434 2
0,05	1030 8	3249	-	1486 8	1221 8	1172 2
0,07 5	7723	-	-	1361 0	6730	1032 2
0,1	-	-	-	1210 5	6084	4088

Таблица 2. Результаты экспериментов решения СЛАУ задачи определения местоположения БПЛА на основе второго способа организации вычислительного процесса

∇t_2	Высота ЛА для дельта-преобразований второго порядка, H_2					
	$V_1 = 700$ м/с			$V_1 = 300$ м/с		
	(8)	(9)	(10)	(8)	(9)	(10)
0,00 5	1670 5	1635 8	1663 2	1693 8	1684 3	1695 1
0,01	1586 8	1288 7	1564 4	1673 8	1661 5	1669 2
0,02 5	9914	6630	9494	1566 7	1544 6	1539 7
0,05	-	-	-	1179 0	9830	1133 2
0,07 5	-	-	-	5424	5705	4433

Анализ полученных в таблицах 1, 2 данных показывает, что при решении СЛАУ на базе дельта-преобразований второго порядка обеспечивается установившийся процесс и решение СЛАУ за одну итерацию уже на достаточно большой высоте по отношению к начальной с представляющим интерес для практического использования временным шагом ∇t_2 . При этом предполагается, что получение результата для всех уравнений системы осуществляется в пределах ~ 10 тактов при реализации алгоритма в специализированном вычислителе на базе ПЛИС. Организация вычислительного процесса как первым, так и вторым способом показывает в значительной мере схожие результаты, однако наблюдается преимущество использования второго способа при расположении маяков в большей близости к поверхности земли.

Для управления БПЛА могут быть взяты вычисляемые координаты после начала установившегося процесса, то есть, начиная с высоты H_2 .

Во втором типе задач (при посадке БПЛА) рассмотрены следующие варианты расположения 4-х маяков:

$$(100, -40, 1), (150, -100, 0), (10, 400, 0), (-15, 450, 10). \quad (11)$$

$$(100, -40, 5), (150, -100, 25), (10, 400, 0),$$

$$(-15, 450, 25). \quad (12)$$

Принимаем также: начальная высота БПЛА $H_0 = 200$ м, скорость м/с, угол к поверхности земли 4° .

В таблице 3 представлены результаты, характеризующие временные шаги ∇t_2 и высоту H_2 в метрах при использовании алгоритма на базе дельта-преобразований второго порядка [7-8] для первого и второго способов организации вычислительного процесса. Параметр H_2 отражает высоту, начиная с которой обеспечивается решение СЛАУ за одну итерацию в установившемся процессе с заданной одинаковой максимальной точностью решения 2^{-13} (характеризующейся значением 0,12 м.) для двух вариантов размещения маяков (11), (12).

Таблица 3. Результаты экспериментов решения СЛАУ задачи определения местоположения БПЛА (при посадке БПЛА).

∇t_2	Высота БПЛА для дельта-преобразований второго порядка, H_2			
	на основе первого способа организации вычислительного процесса		на основе второго способа организации вычислительного процесса	
	(11)	(12)	(11)	(12)
0,005	155	169	192	196
0,01	115	124	183	194
0,025	30	17	148	191
0,05			103	159
0,75				111
0,1				20

Анализ полученных результатов, представленных в таблице 3, показывает, что при решении задачи определения местоположения БПЛА при посадке, вычислительный процесс целесообразно организовать на основе второго способа. При этом первый временной шаг решения реализуется при фиксированных значениях дальностей. После завершения переходного процесса осуществляется решение на очередном временном шаге с реализацией одной итерации. Установившийся процесс начинается с момента использования наименьшего значения кванта преобразования. Для управления БПЛА могут быть использованы вычисляемые координаты после начала установившегося процесса, начиная с высоты H_2 .

В исследованиях рассматривалось решение СЛАУ без учета ошибок в оценке координат расположения маяков и измерения дальностей, вопросы целостности не рассматривались.

Заклучение

В работе рассмотрены особенности алгоритмизации решения задачи определения местоположения БПЛА с использованием четырех маяков при наземной подготовке предложенного задания, в которых описаны все математические преобразования, необходимые для применения итерационных методов в целом, и в частности, для использования методов и алгоритмов на основе дельта-преобразований. Кроме того, в работе представлены особенности решения задачи на борту летательного аппарата в режиме реального времени с использованием дельта-преобразований, позволяющие реализовать высокопроизводительный экономичный по аппаратным ресурсам специализированный вычислитель для одновременного решения большого количества СЛАУ на основе дельта-преобразований второго порядка.

В работе проведено исследование вопросов проектирования высокопроизводительных экономичных по аппаратным ресурсам специализированных вычислителей для одновременного решения большого количества СЛАУ на основе дельта-преобразований второго порядка, которое показало, что использование оптимизированных дельта-преобразований второго порядка представляет возможность решения СЛАУ с переменными свободными членами на каждом временном шаге установившегося процесса за одну итерацию. Полученные результаты экспериментов с использованием компьютерного моделирования подтверждают возможности оперирования с практически значимыми временными шагами работы системы при достаточно большой удаленности начала установившегося процесса от начала координат.

Одним из перспективных направлений дальнейших исследований является исследование возможности эффективной реализации бортовых специализированных средств для решения СЛАУ в определении координат ЛА на основе разработанных методов и алгоритмов с использованием дельта-преобразований второго порядка на основе ПЛИС.

Литература

1. Барабанов О.О., Барабанова Л.П. Математические задачи дальномерной навигации. – М.: ФИЗМАТЛИТ, 2007. – 272 с.
2. Скрыпник, О.Н. Радионавигационные системы воздушных судов: Учебник/ О.Н. Скрыпник. – М.: ИНФРА-М, 2014. – 348 с.
3. Кравченко П.П. Оптимизированные дельта-преобразования второго порядка. Теория и применение. Монография. – М: Радиотехника, 2010. – 288 с.

4. Кравченко П.П., Пирская Л.В. Итерационный метод решения систем линейных алгебраических уравнений, исключаящий операцию многоразрядного умножения // Известия ЮФУ. Технические науки. – 2014. – № 7 (156). – С. 214-224.
5. Pirskeya, L.V. Iterative Algorithm for Solving of Linear Algebraic Equations Systems without Multi-bit Multiplication Operation // Engineering and Telecommunication (EnT), 2014 International Conference on. – IEEE, 2014. – P. 87 – 91.
6. Kravchenko P.P., Pirskeya L.V. The method of organizing the iterative process of the system of the linear algebraic equations solution excluding the multidigit multiplication operation// Biosciences Biotechnology Research Asia December. – 2014. – Vol. 11(3). – P.1831-1839.
7. Кравченко П.П., Пирская Л.В. Метод организации итерационного решения систем линейных алгебраических уравнений с использованием дельта-преобразований второго порядка // Известия ЮФУ. Технические науки. – 2015. – № 6 (167). – С. 57-71.
8. Kravchenko P.P., Pirskeya L.V., Khusainov N. Sh. Algorithm of iterative solution of linear algebraic equations systems based on the second order delta-transformation for specialized computers of real-time systems// Biosciences Biotechnology Research Asia December. – 2015. – Vol. 11(Spl. Edn. 2). – P. 279-289.
9. Березин, И.С. Методы вычислений, том 2. - М.: Наука, 1966. – 632 с.
10. Самарский, А.А. Численные методы. – М.: Наука, 1989. – 432 с.
11. Фаддеев, Д. К. Фаддеева. В.Н. Вычислительные методы линейной алгебры, 4-е изд., стереотип. - СПб.: Лань, 2009. – 734 с.
12. Greenbaum, A. Iterative Methods for Solving Linear Systems. – Philadelphia, PA. SIAM, 1997. – 220 p.
13. Yang H., Ziavras S.G. FPGA-based vector processor for algebraic equation solvers // IEEE International SOC Conference. – IEEE, 2005. – P. 115-116.
14. Zhang W., Betz V., Rose J. Portable and Scalable FPGA-Based Acceleration of a Direct Linear System Solver // ACM Transactions on Reconfigurable Technology and Systems (TRETS). – 2012. – Vol. 5, №1. – Article 6.
15. Хусаинов Н.Ш., Кравченко П.П., Лутай В.Н., Тарасов С.А., Щербинин В.В. Системы радионавигации современных и перспективных летательных аппаратов. Ч. 1. Методы определения местоположения и автономный контроль целостности: Монография. – Таганрог: Изд-во ЮФУ, 2015. – 118 с.
16. Шебшаевич, В.С., Дмитриев П.П., Иванцевич Н.В. и др. Сетевые спутниковые радионавигационные системы. – М.: Радио и

связь, 1993. – 408 с.

Пирская Л.В., Кравченко П.П. Исследование использования дельта-преобразований для определения местоположения беспилотного летательного аппарата. В работе рассматриваются возможности решения задачи определения местоположения беспилотного летательного аппарата (БПЛА) специализированным вычислительным устройством. В постановке задачи рассматривается определение координат БПЛА на основе координат расположенных на относительно близком расстоянии от начала локальной системы координат групп четверок маяков и дальности от БПЛА до маяков. На основании имеющихся данных для каждой группы формируется в стандартном для дальномерной навигации система четырех уравнений, которая преобразовывается к СЛАУ третьего порядка. Решение данной СЛАУ с непрерывными переменными свободными членами определяет координаты БПЛА. В работе рассмотрены особенности алгоритмизации решения задачи определения местоположения БПЛА при наземной подготовке предложенного задания и решении задачи на борту БПЛА в режиме реального времени. Проведено исследование возможности использования дельта-преобразований второго порядка для решения СЛАУ в поставленной задаче на различных примерах размещения маяков. Полученные результаты экспериментов с использованием компьютерного моделирования подтверждают возможность получения решения СЛАУ за одну итерацию в установившемся процессе с временным шагом, представляющим практический интерес, а начало установившегося процесса обеспечивается на достаточно большой удаленности от начала координат.

Ключевые слова: Определение координат БПЛА, решение СЛАУ, дельта-преобразование второго порядка, бортовое специализированное вычислительное устройство.

Pirskaya L.V., Kravchenko P.P. Research on use delta-transformations to determine the position of unmanned aircraft. The paper considers the possibility of solving the problem of determining the location of an unmanned aerial vehicle (UAV) using specialized computing device. The mission statement addresses the determination of the coordinates of UAV based on the coordinates located at a relatively short distance from the start of the local coordinate system of groups of fours beacons and range from the UAV to the beacons. Based on the available data for each group is formed in the standard navigation system rangefinder four equations, which is converted to the third-order linear systems. The solution of this linear systems with continuous variables constant terms determines the coordinates of the UAV. The paper discusses the features of algorithmization solve the problem of determining the location of the UAV ground when preparing the proposed tasks and solving problems on board UAVs in real time. The investigation of the possibility of using second-order delta-transformations for solving linear systems in the task at different examples of beacons placement. The results of experiments using computer simulations confirm the possibility of solving the linear in one iteration in a steady course with a time step of practical concern, and the beginning set the process is provided at a sufficiently large distance from the origin. .

Keywords: Aircraft position finding, linear system solution, second order delta-transformation, aircraft special-purpose calculating unit.

Статья поступила в редакцию 20.11.2016

Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.415.532.3

Формальная верификация циклических программ

О.И.Федяев

Донецкий национальный технический университет
fedyaev@donntu.org

Федяев О.И. *Формальная верификация циклических программ.* В статье рассматривается аналитический подход к оценке корректности сложных программ. На примере доказательства правильности программы сортировки проведена апробация теоретико-функционального метода верификации. Формализм программных функций позволил математически строго преодолеть сложности доказательства правильности обработки элементов массива во вложенных циклах. Результаты статьи подтверждают возможность применения данного метода на практике в программной инженерии.

Ключевые слова: верификация программы, корректность программы, программная функция, доказательство правильности программы, теоретико-функциональный подход.

Введение

Важную роль в обеспечении качества создаваемого программного продукта играет формальная верификация программ [1,2]. Благодаря математическому доказательству корректности программного кода, т.е. его соответствия спецификации решаемой задачи, верификация повышает уровень логической строгости инженерии программного обеспечения [3,4,5].

Среди существующих подходов к формальной верификации выделяется своей оригинальностью и чёткой логикой теоретико-функциональный метод доказательства правильности структурированных программ, который может успешно применяться для проверки корректности циклических программ, использующих массивы [6].

В этом методе правильность программы определяется как соответствие между программой P и её заданной функцией f . Алгебраическая структура исходной программы P характеризует её как составную, что даёт возможность декомпозировать P на элементарные составляющие. Это позволяет задачу верификации программы P свести на основании аксиомы замещения к проверке правильности элементарных подпрограмм P_i , из которых она состоит. Верификация правильности элементарных программ без циклов осуществляется с помощью анализа соответствующих E - схем. Лемма о переходе от итеративных программ к рекурсивным позволяет свести задачу верификации циклических базовых программ P_i (типа while, until, for) к задаче верификации функционально эквивалентных программ без циклов. Для вывода программных функций $[P_i]$ используются трассировочные таблицы и разделяющиеся условные правила. В данном методе проблема перестановки элементов

решается детальным развёртыванием и свёртыванием изменяющихся частей массива. Формально полная правильность программы P состоит в нахождении программной функции $[P]$ и сравнении её с заданной функцией f , т.е. в проверке равенства $f = [P]$.

В данной работе ставится задача оценить эффективность данного метода на примере формальной верификации программы сортировки по убыванию элементов массива методом выбора [7]. Алгоритм сортировки на рис.1 составлен специально так, чтобы в цикле присутствовали перестановки элементов массива, что делает задачу верификации не тривиальной. Процесс верификации программы P включает последовательность шагов доказательства правильности элементарных программ, из которых состоит P .

Шаг 1. Верификация подпрограммы типа «if-then-else»

В блок-схеме алгоритма сортировки на рис.1 первой элементарной программой (обозначим её P_1), с которой надо начать проверку правильности, является «внутренний» условный оператор (рис.2).

Поскольку функция f_1 для программы P_1 не задана, то выдвинем гипотезу о ней и запишем её в виде следующего предложения одновременного присваивания [6]

$$f_1 = (i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i)) \quad (1)$$

Эта запись означает, что в начале вычисляются значения всех выражений, стоящих справа от символа «:=», а затем полученные значения присваиваются соответствующим именам данных, стоящим слева. Правильность P_1 следует из доказательства равенства $f_1 = [P_1]$. Поэтому найдём программную функцию $[P_1]$ с помощью трассировочной таблицы.

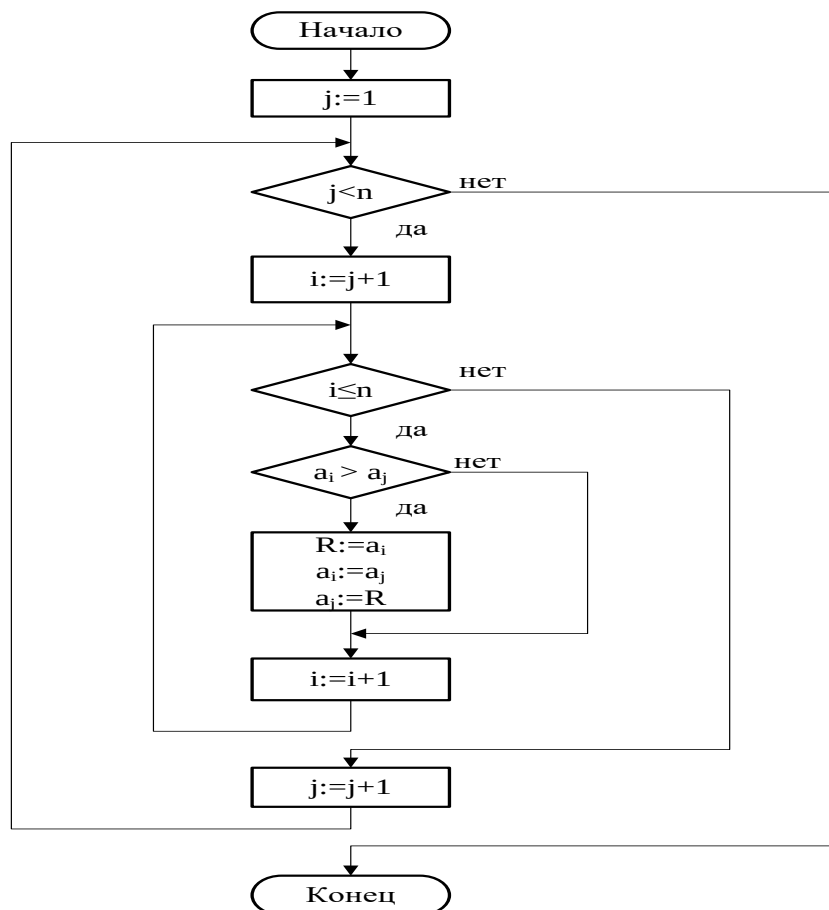
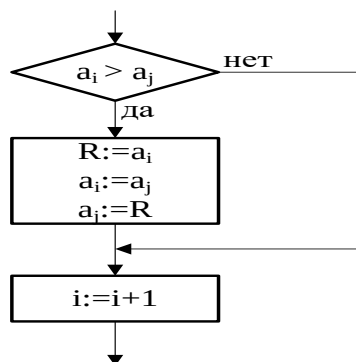


Рисунок 1 - Блок-схема программы P сортировки массива

Рисунок 2 - Элементарная программа P₁ типа «if-then-else»

Трассировочная таблица позволяет формально записать вывод функции условно последовательной программы P₁. Трассировка

первого пути выполнения программы P₁ показана в табл. 1.

Таблица 1. Трассировка пути выполнения P₁, когда $(a_i > a_j) = \text{true}$

N п/п	Фрагмент	a_j	a_i	R	i
1	$a_i > a_j$	$a_j^1 = a_j^0$	$a_i^1 = a_i^0$	$R^1 = R^0$	$i^1 = i^0$
2	$R := a_i$	$a_j^2 = a_j^1$	$a_i^2 = a_i^1$	$R^2 = a_i^1$	$i^2 = i^1$
3	$a_i := a_j$	$a_j^3 = a_j^2$	$a_i^3 = a_j^2$	$R^3 = R^2$	$i^3 = i^2$
4	$a_j := R$	$a_j^4 = R^3$	$a_i^4 = a_i^3$	$R^4 = R^3$	$i^4 = i^3$
5	$i := i + 1$	$a_j^5 = a_j^4$	$a_i^5 = a_i^4$	$R^5 = R^4$	$i^5 = i^4 + 1$

Если просматривать снизу вверх соответствующие колонки таблицы и при этом систематически исключать все промежуточные значения верхних индексов, начиная с конечных значений, то получим программную функцию для первого пути, исключив побочное данные R:

$$\begin{aligned} & - \text{изменение данных} \\ & a_j^5 = a_j^4 = R^3 = R^2 = a_i^1 = a_i^0 ; \\ & a_i^5 = a_i^4 = a_i^3 = a_j^2 = a_j^1 = a_j^0 ; i^5 = i^4 + 1 = \\ & = i^3 + 1 = i^0 + 1 ; \end{aligned}$$

$$\begin{aligned} & - \text{программная функция пути} \\ & (a_i > a_j) \rightarrow i, a_j, a_i := i + 1, a_i, a_j . \quad (2) \end{aligned}$$

Выполнив аналогично трассировку второго пути выполнения программы P_1 , когда $(a_i > a_j = \text{false})$, получим другую часть программной функции, соответствующую второму пути:

$$(a_i \leq a_j) \rightarrow i, a_j, a_i := i + 1, a_j, a_i . \quad (3)$$

Объединение (2) и (3) даёт общую программную функцию $[P_1]$ в виде условного правила

$$[P_1] = (a_i > a_j) \rightarrow i, a_j, a_i := i + 1, a_i, a_j \mid$$

$$(a_i \leq a_j) \rightarrow i, a_j, a_i := i + 1, a_j, a_i ,$$

которое можно записать в более компактной форме

$$[P_1] = (i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i)) . \quad (4)$$

Сравнивая (1) и (4) видно, что $f_1 = [P_1]$ и, следовательно, высказанная гипотеза о программной функции верна.

Шаг 2. Верификация подпрограммы типа «while»

На основании аксиомы замещения [6] можно выполнить замену подпрограммы P_1 , правильность которой доказана на шаге 1, одним оператором с той же программной функцией f_1 . При этом программная функция исходной программы P не изменится. После такой подстановки выделяется следующий элементарный структурный элемент для верификации, который является циклом типа «while» (рис. 3).

Как видно из рисунка, тело цикла представлено одним оператором, который реализует функцию f_1 . Исходная функция, которую должна реализовать верифицируемая программа P_1 , не задана. Поэтому сначала сформулируем гипотезу о функции для элементарной циклической программы типа **while** (рис. 3). Для этого достаточно рассмотреть три шага работы оператора цикла, начиная с номера $i=j+1$.

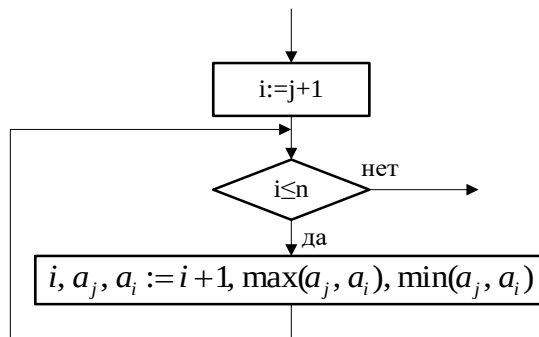


Рисунок 3 - Элементарная программа P_2 типа «while»

Перестановка элементов в группе из четырёх последовательных элементов массива



будет определяться следующими операторами одновременного присваивания, которые показывают, как изменяются значения элементов

в этих позициях на каждом цикле в зависимости от исходных значений элементов перед началом циклического процесса:

$$1 \text{ шаг: } a_j^1, a_i^1, a_{i+1}^1, a_{i+2}^1 := \max(a_j, a_i), \min(a_j, a_i), a_{i+1}, a_{i+2} ;$$

$$2 \text{ шаг: } a_j^2, a_i^2, a_{i+1}^2, a_{i+2}^2 := \max(\max(a_j, a_i), a_{i+1}), \min(a_j, a_i), \min(\max(a_j, a_i), a_{i+1}), a_{i+2} ;$$

$$3 \text{ шаг: } a_j^3, a_i^3, a_{i+1}^3, a_{i+2}^3 := \max(\max(\max(a_j, a_i), a_{i+1}), a_{i+2}), \min(a_j, a_i), \min(\max(a_j, a_i), a_{i+1}), \min(\max(\max(a_j, a_i), a_{i+1}), a_{i+2}) .$$

Из этих шагов видна закономерность, которую можно записать в виде следующей функции:

$$f_2 = (j < i < n) \rightarrow i, a_j, a(i:n) := n + 1, \max \left[a_j, \max[a(i:n)] \right],$$

$$\boxed{i} \quad \min[a_i, a_j],$$

$$\boxed{i+1} \quad \min[a_{i+1}, \max[a_j, a(i:i)]]],$$

$$\boxed{i+2} \quad \min[a_{i+2}, \max[a_j, a(i:i+1)]]],$$

$$\dots$$

$$\boxed{n} \quad \min[a_n, \max[a_j, a(i:n-1)]]].$$

В прямоугольных рамках указаны номера элементов массива. Теперь докажем правильность следующей циклической программы

$P_2 = \mathbf{while} \ i \leq n \ \mathbf{do} \ i, a_j, a_i :=$

$i + 1, \max(a_j, a_i), \min(a_j, a_i) \ \mathbf{od} \ ,$

$P'_2 = \mathbf{if} \ i < n$

$\mathbf{then} \ i, a_j, a_i := i + 1, \max(a_j, a_i), \min(a_j, a_i);$

$(j < i < n) \rightarrow i, a_j, a(i:n) := n + 1, \max[a_j, \max[a(i:n)]]],$

$\min[a_i, a_j],$

$\min[a_{i+1}, \max[a_j, a(i:i)]]],$

$\min[a_{i+2}, \max[a_j, a(i:i+1)]]],$

$\dots$

$\min[a_n, \max[a_j, a(i:n-1)]]];$

$\mathbf{fi} \ .$

где ключевые слова **do** и **od** являются обычными ограничителями фрагментов программы.

Используя лемму о рекурсивном представлении [6], заменим программу P_2 на рекурсивную нециклическую программу P'_2 , которая будет иметь следующий вид:

Докажем справедливость равенства $f_2 = [P'_2]$, что эквивалентно доказательству $f_2 = [P_2]$. Найдём с помощью трассировочной таблицы программную

функцию $[P'_2]$.

Таблица 2. Трассировочная таблица для программы P'_2

Фрагмент программы	Условие	Массив a	i
1	$i_0 \leq n$	$a_1(1:n) = a_0(1:n)$	$i_1 = i_0$
2	$j < i_1 < n$	(2.1) $a_2(1:j-1) = a_1(1:j-1)$ (2.2) $a_2(j) = \max[a_1(j), a_1(i_1)]$ (2.3) $a_2(j+1:i_1-1) = a_1(j+1:i_1-1)$ (2.4) $a_2(i_1) = \min[a_1(j), a_1(i_1)]$ (2.5) $a_2(i_1+1:n) = a_1(i_1+1:n)$	$i_2 = i_1 + 1$
3	$j < i_2 < n$	(3.1) $a_3(1:j-1) = a_2(1:j-1)$ (3.2) $a_3(j) = \max[a_2(j), \max[a_2(i_2:n)]]$ (3.3) $a_3(j+1:i_2-1) = a_2(j+1:i_2-1)$ (3.4) $a_3(i_2) = \min[a_2(i_2), a_2(j)]$ (3.5) $a_3(i_2+1) =$ $\min[a_2(i_2+1), \max[a_2(j), a_2(i_2:i_2)]]$ $\dots$ $a_3(n) = \min[a_2(n), \max[a_2(j), a_2(i_2:n-1)]]$	$i_3 = n + 1$

Условие:

$$\begin{aligned}(i_0 \leq n) \wedge (j < i_1) \wedge (i_1 < n) \wedge (j < i_2) \wedge (i_2 < n) = \\(i_0 \leq n) \wedge (j < i_0) \wedge (i_0 < n) \wedge (j < i_0 + 1) \wedge (i_0 + 1 < n) = \\(j < i_0 \leq n) \wedge (j < i_0 + 1 < n) = (j < i_0 < n)\end{aligned}$$

Если проследить за изменением данных по трассировочной таблице путём систематического исключения всех промежуточных значений

индексов, то получим функцию рекурсивной программы. Из формулы (3.1) в трассировочной таблице (см. табл. 2) выводим:

$$a_3(1:j-1) = a_2(1:j-1) = a_1(1:j-1) = a_0(1:j-1)$$

$$a_3(1:j-1) = a_2(1:j-1) = a_1(1:j-1) = a_0(1:j-1)$$

Таким же способом из формулы (3.2) получаем:

$$\begin{aligned}a_3(j) &= \max[\max[a_1(j), a_1(i_1)], \max[a_2(i_1+1:n)]] \\&= \max[\max[a_0(j), a_0(i_0)], \max[a_1(i_0+1:n)]] \\&= \max[\max[a_0(j), a_0(i_0)], \max[a_0(i_0+1:n)]] \\&= \max[a_0(j), \max[a_0(i_0:n)]]\end{aligned}$$

Из формулы (3.3) получаем:

$$\begin{aligned}a_3(j+1:i_2-1) &= a_2(j+1:i_2-1) \\a_3(j+1:i_1+1-1) &= a_2(j+1:i_1+1-1) \\a_3(j+1:i_1) &= a_2(j+1:i_1) = a_1(j+1:i_1), \min[a_1(j), a_1(i_1)] \\a_3(j+1:i_0) &= a_2(j+1:i_0) = a_0(j+1:i_0-1), \min[a_0(j), a_0(i_0)] \\a_3(j+1:i_0-1) &= a_0(j+1:i_0-1) \\a_3(i_0) &= \min[a_0(j), a_0(i_0)]\end{aligned}$$

Из формулы (3.4) получаем:

$$\begin{aligned}a_3(i_2) &= \min[a_2(i_2), a_2(j)] \\a_3(i_1+1) &= \min[a_2(i_1+1), a_2(j)] \\a_3(i_0+1) &= \min[a_1(i_1+1), \max[a_1(j), a_1(i_1)]] \\&= \min[a_0(i_0+1), \max[a_0(j), a_0(i_0)]] \\&= \min[a_0(i_0+1), \max[a_0(j), a_0(i_0:i_0)]]\end{aligned}$$

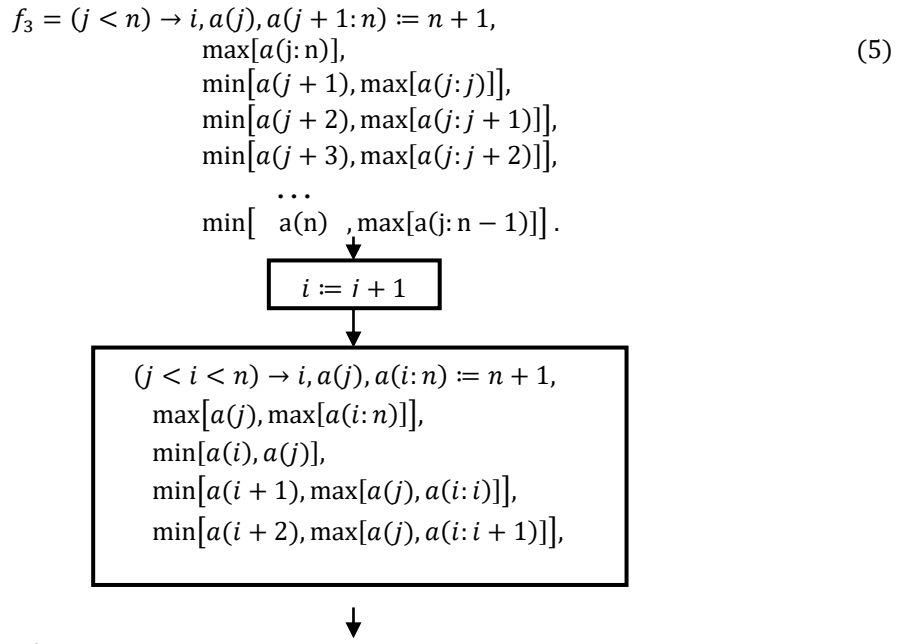
Из формулы (3.5) получаем:

$$\begin{aligned}a_3(i_2+1) &= \min[a_2(i_2+1), \max[a_2(j), a_2(i_2:i_2)]] \\a_3(i_0+2) &= \min[a_2(i_1+2), \max[a_2(j), a_2(i_1+1:i_1+1)]] \\&= \min[a_1(i_1+2), \max[\max[a_1(j), a_1(i_1)], a_2(i_1+1:i_1+1)]] \\&= \min[a_0(i_0+2), \max[a_0(j), a_0(i_0:i_0+1)]] \quad \text{и т. д.}\end{aligned}$$

Если объединить результаты выводов по формулам (3.1) – (3.5) из табл. 2 и отбросить нулевые индексы, то получим программную функцию рекурсивной программы P'_2 . Она полностью совпадает с выдвинутой гипотезой f_2 , т.е. $f_2 = [P'_2]$. Принимая во внимание лемму о рекурсивном представлении программы и теорему правильности [3], можно утверждать, что циклическая программа P_2 также правильна.

Шаг 3. Верификация программы типа «последовательность»

Далее переходим к доказательству правильности следующей элементарной программы P_3 (последовательность), которая получается после применения аксиомы замещения правильного цикла **while** его функцией (рис.4). Выдвинем следующую гипотезу о функции программы P_3 :

Рис. 4. Элементарная программа P_3 типа «последовательность»

Упростим запись функции f_3 , представив её в виде

$$f_3 = (j < n) \rightarrow i, a(j), a(j+1:n) := n+1, \max[a(j:n)], a(j+1:n) \setminus a(j).$$

Символ «\» обозначает операцию исключения элементов. Поэтому запись $a(k:n) \setminus a(j:k)$ означает, что в массиве $a(j:n)$ среди элементов $a(k:n)$ нет элементов этого же массива, стоящих на позициях от j до k ($j < k < n$). Такое расположение элементов массива возникает после перестановки максимальных элементов, т.е. максимальные элементы «всплыли» в левую часть массива $a(j:k)$, а меньшие элементы переместились в позиции $a(k:n)$.

Представленную на рис. 4 последовательность из двух операторов можно рассматривать как трассировочную таблицу. В этом случае, если во втором операторе блок-схемы (т.е. в функции f_2)

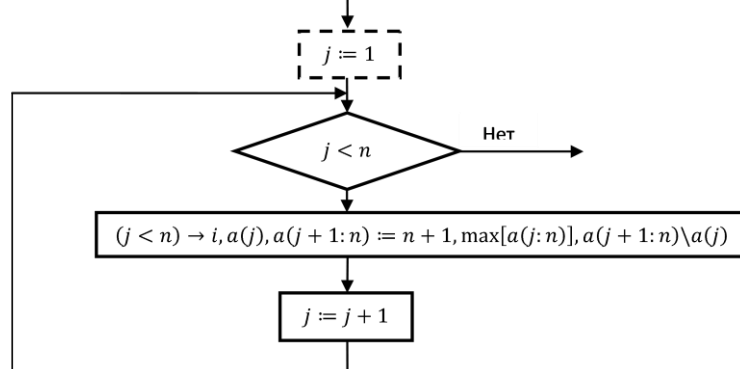
заменить i на $j+1$, то получится программная функция $[P_3]$, которая будет совпадать с f_3 . Таким образом, из равенства $f_3 = [P_3]$ следует правильность программы P_3 .

Шаг 4. Верификация подпрограммы типа «while»

Последующая декомпозиция структуры исходной программы P даёт следующий простой фрагмент для верификации – цикл типа «while», представленный на рис. 5. Тело этого цикла получено на основании аксиомы замещения, путём замены правильной последовательности P_3 на функцию f_3 .

Сформулируем гипотезу для программы P_4 в виде функции f_4 :

$$(j < n) \rightarrow j, i, a(j:n) := n, n+1, \max[a(j:n)], \max[a(j+1:n) \setminus a(j)], \max[a(j+2:n) \setminus a(j:j+1)], \max[a(j+3:n) \setminus a(j:j+2)], \dots, \max[a(n:n) \setminus a(j:n-1)] . \quad (6)$$

Рисунок 5 - Элементарная программа P_4 типа «while»

$$P_4 = \textbf{while } j < n \textbf{ do } (j < n) \rightarrow i, a(j), a(j+1:n) := n+1, \\ \max[a(j:n)], a(j+1:n) \setminus a(j); \\ j := j+1;$$

od;

Далее, используя лемму о рекурсивном представлении [6], представим программу P_4 в рекурсивной ациклической форме P'_4 , что

упростит получение программной функции и саму верификацию:

$$P'_4 = \textbf{if } j < n \textbf{ then } (j < n) \rightarrow i, a(j), a(j+1:n) := n+1, \\ \max[a(j:n)], \\ a(j+1:n) \setminus a(j); \\ j := j+1; \\ (j < n) \rightarrow j, i, a(j:n) := n, n+1, \max[a(j:n)], \\ \max[a(j+1:n) \setminus a(j:j)], \\ \max[a(j+2:n) \setminus a(j:j+1)], \\ \dots \\ \max[a(n:n) \setminus a(j:n-1)];$$

fi

Докажем, что $f_4 = [P'_4]$. Для этого выведем из трассировочной табл.3 программную функцию $[P'_4]$.

Таблица 3. Трассировка штатного пути выполнения программы P'_4

Фрагмент программы	Условие	Массив $a(1:n)$	j
1	$j_0 < n$	$a_1(1:n) = a_0(1:n)$	$j_1 = j_0$
2	$j_1 < n$	(2.1) $a_2(1:j_1-1) = a_1(1:j_1-1)$ (2.2) $a_2(j_1) = \max[a_1(j_1:n)]$ (2.3) $a_2(j_1+1:n) = a_1(j_1+1:n) \setminus a(j_1)$	$j_2 = j_1$
3		$a_3(1:n) = a_2(1:n)$	$j_3 = j_2 + 1$
4	$j_3 < n$	(4.1) $a_4(1:j_3-1) = a_3(1:j_3-1)$ (4.2) $a_4(j_3) = \max[a_3(j_3:n)]$ (4.3) $a_4(j_3+1) = \max[a_3(j_3+1:n) \setminus a(j_3:j_3)]$ (4.4) $a_4(j_3+2) = \max[a_3(j_3+2:n) \setminus a(j_3:j_3+1)]$ $\dots$ $a_4(n) = \max[a_3(n:n) \setminus a(j_3:n-1)]$	$j_4 = n$

Проследим только за изменением элементов массива a и параметра j :

$$\begin{aligned} a_4(1:j_3-1) &= a_3(1:j_3-1) \\ a_4(1:j_2) &= a_3(1:j_2) \\ a_4(1:j_0) &= a_2(1:j_2) = a_2(1:j_1) = a_1(1:j_1-1), \max[a_1(j_1:n)] \\ &= a_0(1:j_0-1), \max[a_0(j_0:n)] \end{aligned}$$

$$\begin{aligned} a_4(j_3) &= \max[a_3(j_3:n)] \\ a_4(j_2+1) &= \max[a_2(j_2+1:n)] = \max[a_2(j_1+1:n)] \\ a_4(j_1+1) &= \max[a_1(j_1+1:n) \setminus a(j_1)] \\ a_4(j_0+1) &= \max[a_0(j_0+1:n) \setminus a(j_0)] \end{aligned}$$

$$\begin{aligned} a_4(j_3+1) &= \max[a_3(j_3+1:n) \setminus a(j_3:j_3)] \\ a_4(j_2+2) &= \max[a_2(j_2+2:n) \setminus a(j_2+1:j_2+1)] \\ a_4(j_1+2) &= \max[a_2(j_1+2:n) \setminus a(j_1+1:j_1+1)] \\ &= \max[a_1(j_1+2:n) \setminus a(j_1) \setminus a(j_1+1:j_1+1)] \\ a_4(j_0+2) &= \max[a_0(j_0+2:n) \setminus a(j_1:j_1+1)] \\ &= \max[a_0(j_0+2:n) \setminus a(j_0:j_0+1)] \end{aligned}$$

$\dots$

$$\begin{aligned}
a_4(n) &= \max[a_3(n:n) \setminus a(j_3:n-1)] \\
&= \max[a_2(n:n) \setminus a(j_2+1:n-1)] \\
&= \max[a_1(n:n) \setminus a(j_1) \setminus a(j_1+1:n-1)] \\
&= \max[a_0(n:n) \setminus a(j_1:n-1)] \\
&= \max[a_0(n:n) \setminus a(j_0:n-1)] \\
j_4 &= n.
\end{aligned}$$

Из анализа изменения элементов массива видно, что перемещались только те элементы, которые расположены после j -го номера. Причём, функционально изменение полностью совпадают с выдвинутой гипотезой. Если рассмотреть также и другой путь выполнения программы P_4 , связанный с ложностью выполнения условия $(j_3 < n) = \text{false}$, то совпадают будут не только данные, но и предикат программной функции. Отсюда следует выполнение равенства $f_4 = [P_4']$ и правильность цикла P_4 .

Шаг 5. Верификация программы типа «последовательность»

На последнем этапе процесс верификации сводится к доказательству правильности элементарной программы (рис.6), полученной путём свёртки по аксиоме замещения последовательного ряда доказанных ранее правильных подпрограмм. Гипотезу о программной функции f_5 , реализуемую программой P_5 , представим в следующем виде:

$$\begin{aligned}
f_5 = (1 < n) \rightarrow j, i, a(1:n) := n, n+1, \max[a(1:n)], \\
\max[a(2:n) \setminus a(1)], \\
\max[a(3:n) \setminus a(1:2)], \\
\vdots \\
\max[a(n:n) \setminus a(1:n-1)].
\end{aligned}$$

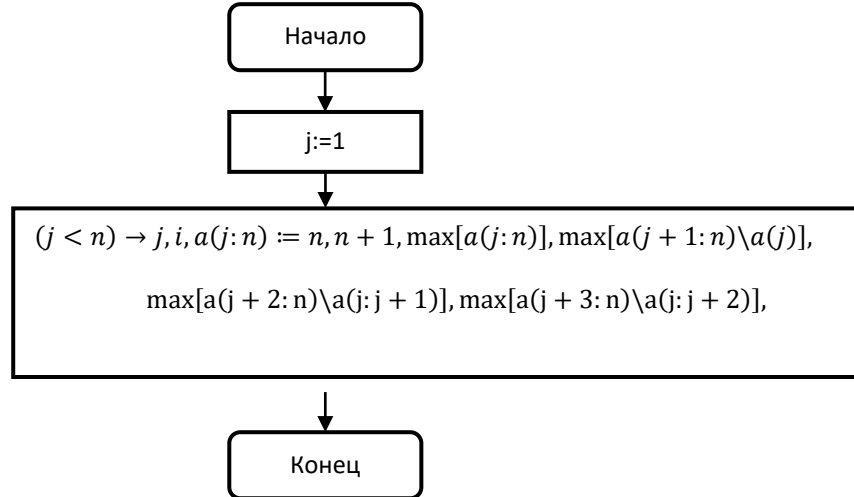


Рисунок 6 - Элементарная программа P_5 типа «последовательность»

Последовательность операторов программы, приведенная на рис.6, позволяет легко получить программную функцию $[P_5]$ без трассировочной таблицы. Для этого достаточно во втором операторе (функции f_4) заменить

Поскольку значения функции $\max$ удовлетворяют следующему неравенству

$$\max[a(1:n)] \geq \max[a(2:n) \setminus a(1)] \geq \max[a(3:n) \setminus a(1:2)] \geq \dots,$$

то можно утверждать, что элементы массива $a(1:n)$ отсортированы по убыванию правильно, т.е. справедливо $a(1:n) = \text{SORT}(a(1:n))$.

переменную j на константу 1. Полученная таким образом программная функция полностью совпадает с гипотезой, т.е. $f_5 = [P_5]$, что подтверждает правильность P_5 и, следовательно, правильность исходной программы P .

Заключение

Возрастающая сложность и важность

разрабатываемых программ требует серьёзного отношения к обеспечению качества производимого программного продукта.

По этой причине в последние годы сделаны огромные усилия по созданию новых технологий разработки программного обеспечения (ПО), методов и инструментальных средств анализа корректности ПО [8,9]. Традиционное тестирование, опираясь на разнообразные методы и мощные средства автоматизации, однако, не могут исключить применение формальных методов доказательства корректности программ, особенно – критически важных. Наиболее перспективный путь в решении проблемы качества ПО заключается в сочетании методов тестирования и верификации программ в составе специализированных инструментальных CASE-систем [10].

В данной работе выполнена апробация формального метода верификации, основанного на понятии программной функции. На примере верификации достаточно сложной циклической программы обработки массива показана практическая возможность применения данного метода в программной инженерии.

Литература

1. IEEE 1012-2004. Standard for Software Verification and Validation. IEEE, 2005.
2. IEEE 1059-1993. Guide for Software Verification and Validation Plans. New York: IEEE, 1993.

3. Непомнящий В.А., Рякин О.М. Прикладные методы верификации программ / Под ред. А.П. Ершова. – М.: Радио и связь, 1988. – 256 с.

4. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ: Model Checking. Пер. с англ./ Под ред. Р.Смелянского. – М.: МЦНМО, 2002. – 416 с.

5. Вудкок Д. Первые шаги к решению проблемы верификации программ. Открытые системы. – 2006. – № 8. – С. 36-43.

6. Лингер Р. И., Миллс Х., Уитт Б. Теория и практика структурного программирования: Пер. с англ. М.: Мир, 1982. – 406 с.

7. Седжвик Р. Фундаментальные алгоритмы на C++. Анализ, структуры данных, сортировка, поиск: Пер. с англ./ Р. Седжвик. – СПб.: ООО «ДиаСофтЮП», 2002. – 688 с.

8. Beckert B., Hahnle R., Schmitt P.H., eds. Verification of Object-Oriented Software: The KeY Approach. Springer, 2007.

9. Hoare T., Misra J. Verified software: Theories, Tools, Experiments. Vision of Grant Challenge project. Microsoft Research Ltd and the University of Texas at Austin, 2005. – P. 1–43.

10. Васенин В.А. Кривчиков М.А. Языково-ориентированное программирование для формальной верификации программного обеспечения. Материалы четвертой Научно-практической конференции «Актуальные проблемы системной и программной инженерии». Сб. науч. тр. / Национальный исследовательский университет «Высшая школа экономики». – М.: Изд-во НИУ ВШЭ, 2015. – 234 с

Федяев О.И. Формальная верификация циклических программ. В статье рассматривается аналитический подход к оценке корректности сложных программ. На примере доказательства правильности программы сортировки проведена апробация теоретико-функционального метода верификации. Формализм программных функций позволил математически строго преодолеть сложности доказательства правильности обработки элементов массива во вложенных циклах. Результаты статьи подтверждают возможность применения данного метода на практике в программной инженерии.

Ключевые слова: верификация программы, корректность программы, программная функция, доказательство правильности программы, теоретико-функциональный подход.

Fedyayev O. Formal verification cycle programs. The article deals with an analytical approach to the assessment of the correctness of complex programs. On the example of the proof of the correctness of the program sorting is carried out check method of the program functions. Formalisms of the program functions allow a mathematically rigorous proof the correctness perform nested loops with arrays. Article results confirm the possibility of using this method in practice in software engineering.

Keywords: Verification of the program, program correctness, program function, proof of the correctness of the program, method of the program functions.

Статья поступила в редакцию 20.11.2016

Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.0-519.854.001

Компьютерное исследование и визуализация колебательной динамики систем в комплексной форме

Филер З.Е.¹, Андрухин А.И.²¹Кировоградский национальный педагогический университет,²Донецкий национальный технический университет

filer@ramber.ru, alexandruckin@ramber.ru

Филер З.Е., Андрухин А.И. Компьютерное исследование и визуализация колебательной динамики систем в комплексной форме. Рассматриваются колебания с помощью использования комплексной формы уравнения. Выделяются линейные части. Выполняется их последующее интегрирование. Выполняется. Фазовые портреты и графики перемещений строятся с использованием пакета Вольфрам Математика. Результаты расчетов приведены.

Ключевые слова: колебания, нелинейные, комплексные уравнения

Введение

В теории колебаний [1] используются комплексные переменные как аппарат анализа решений благодаря формуле Эйлера $e^{i\alpha t} = \cos(\alpha t) + i \sin(\alpha t)$. Нам неизвестны факты широкого использования дифференциальных уравнений в комплексной форме для получения решений и построения их графиков.

В настоящее время существуют пакеты, такие как Wolfram Mathematica, которые позволяют строить их решения и соответствующие графики, используя их комплексную форму.

Она позволяет снизить порядок уравнений вдвое и упростить программирование формул для получения решений и построения их графиков.

1 Постановка задачи

Уравнения автономных колебаний

$$\ddot{x} + f(x) = 0, f(0) = 0, f'(0) = \omega^2 \quad (1)$$

может быть заменено системой уравнений

$$\begin{aligned} \dot{x} &= -\omega y, \\ \dot{y} &= \omega x + (f(x) - \omega^2 x) / \omega, \end{aligned} \quad (2)$$

Последняя формула, может быть заменена в свою очередь, уравнением

$$\frac{dy}{dx} = \frac{-\omega^2 x + (\omega^2 x - f(x))}{\omega^2 y}. \quad (3)$$

Можно перейти от системы (2) к одному уравнению для комплексной функции $z(t) = x(t) + iy(t)$:

$$\dot{z} = i\omega z + i/\omega (f(\operatorname{Re} z) - \omega^2 \operatorname{Re} z). \quad (4)$$

Второе слагаемое в (4) не содержит постоянной и линейной слагаемых в составе правой части. Если же функция $f(x)$ линейна, то

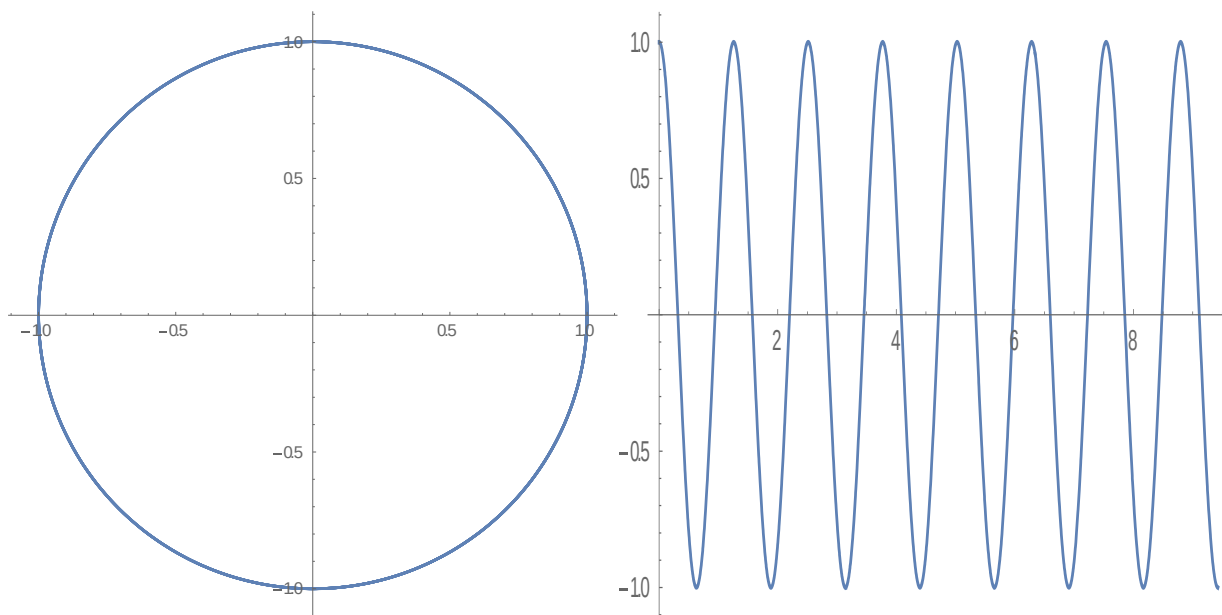
$$\begin{aligned} z &= z_0 e^{i\alpha t} \Rightarrow x = a \cos(\alpha t) \text{ при} \\ z_0 &= a \in R. \end{aligned} \quad (5)$$

Очевидно, $\omega = \sqrt{f'(0)}$ – частота малых колебаний, а поэтому период малых колебаний $T(0) = 2\pi / \omega$.

Между прочим, линейное уравнение первого порядка

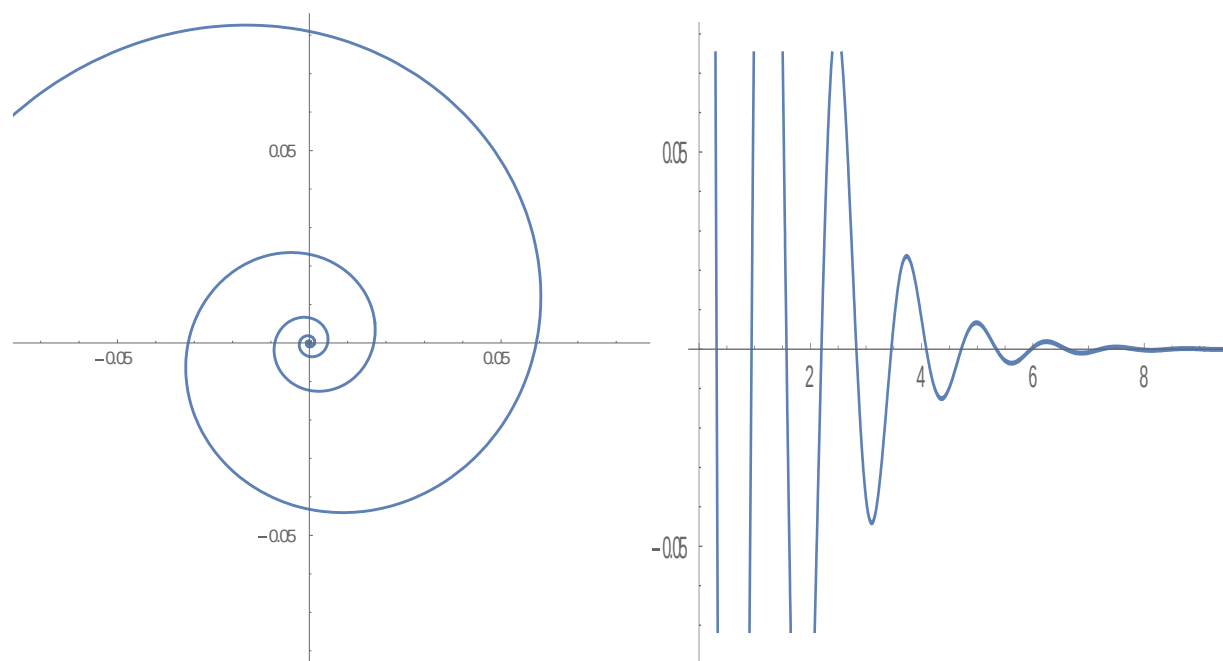
$$\dot{z} = i\omega z + (a + ib)z \quad (6)$$

имеет решение $z(t) = z_0 e^{at} \cdot e^{i(\omega+b)t}$ и при $a \neq 0$ имеет фокус (неустойчивый при $a > 0$ и устойчивый при $a < 0$ (рис. 1).

Рисунок 1а. — Незатухающие колебания при $a=0$

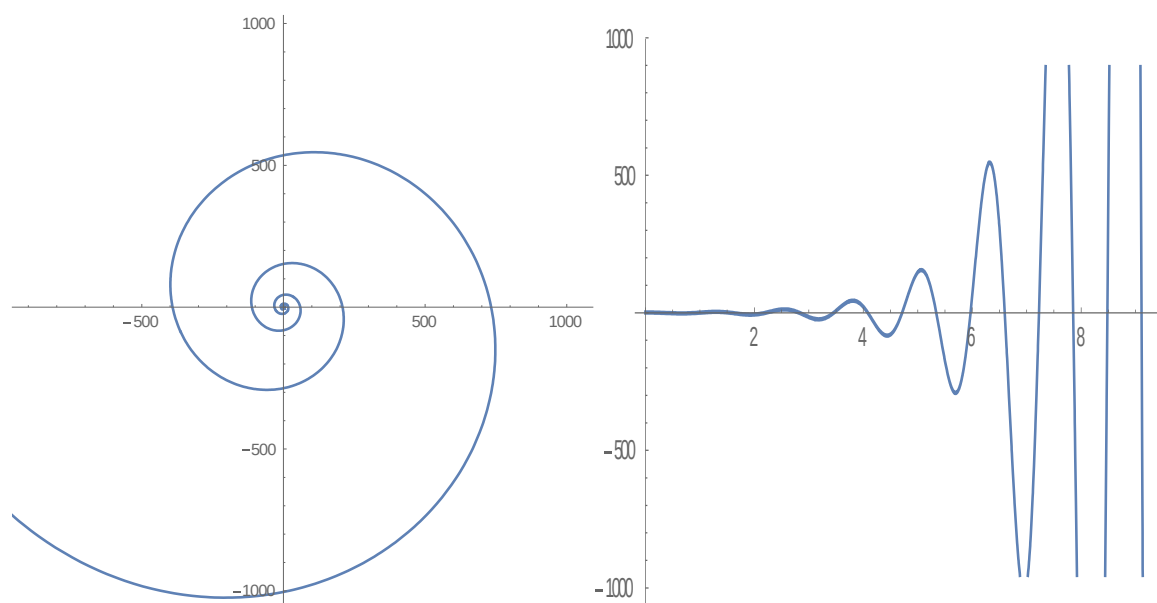
На рис.1а $a=0$, $b=2$ и $\omega=3$, $z_0=1$ и меньше. Построим графики $z(t)$ и $\text{Re}(z(t))$. Первый график будет близок к окружности при

$0 \leq t \leq \frac{2\pi}{5}$, второй – к косинусоиде амплитудой 1 и того же периода.

Рисунок 1б. — Затухающие колебания при $a=-1$.

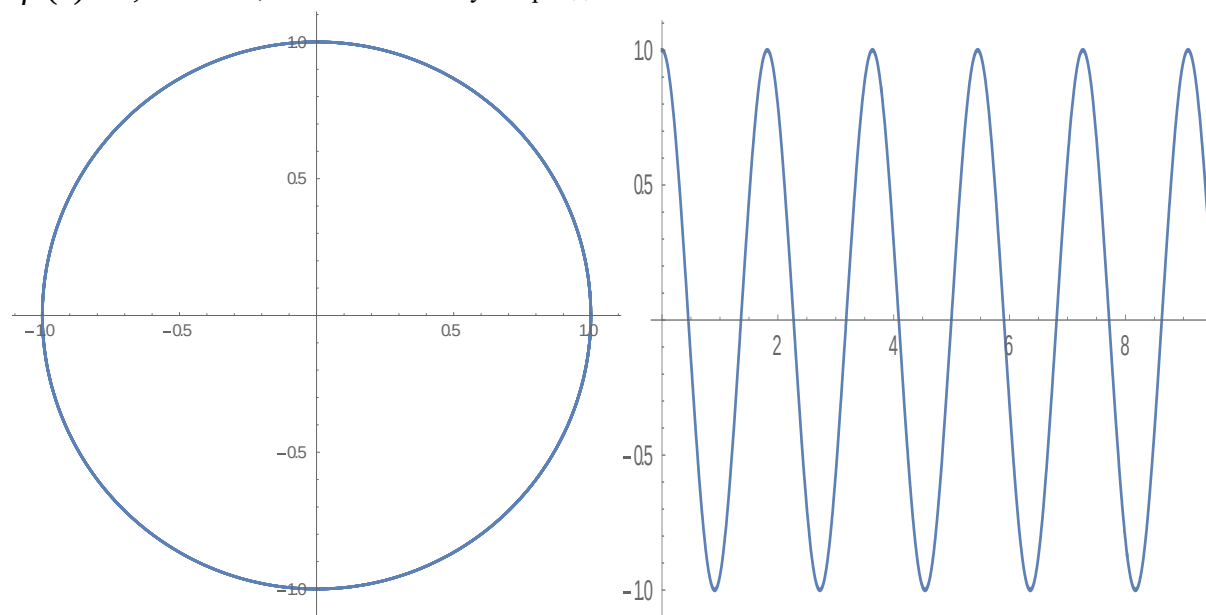
Второй рис.1б при $a=-1$ и остальных данных – закручивающаяся спираль и затухающая косинусоида; Третий график рис.1в

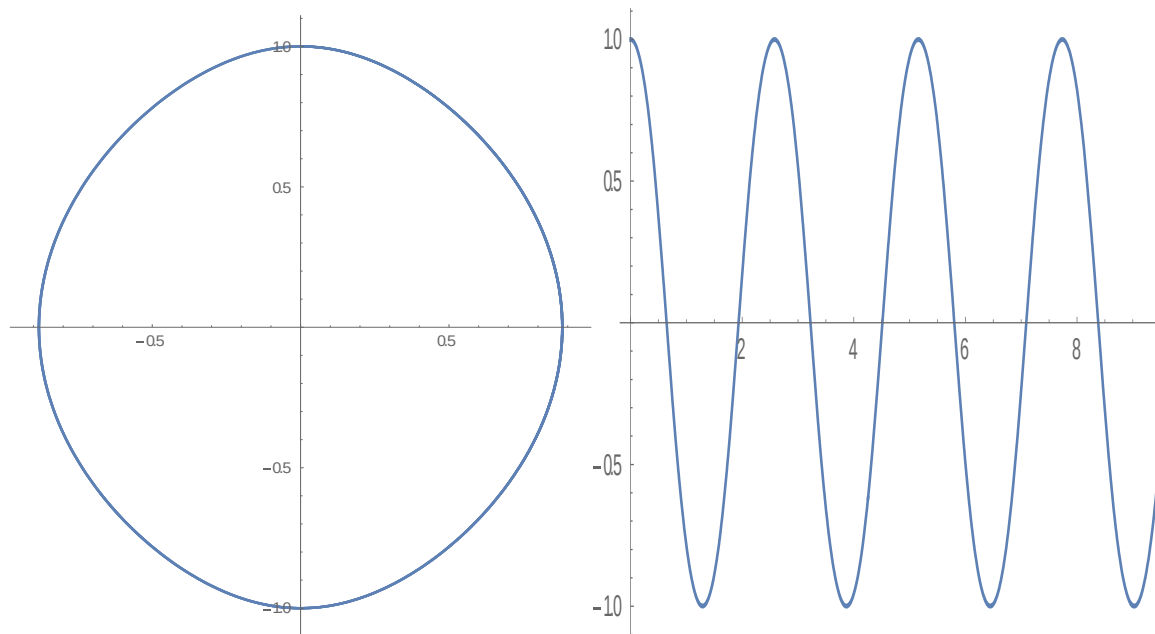
будет при $a=1$ – раскручивающаяся спираль и нарастающая косинусоида.

Рисунок 1в. — Нарастающие колебания при $a=+1$.

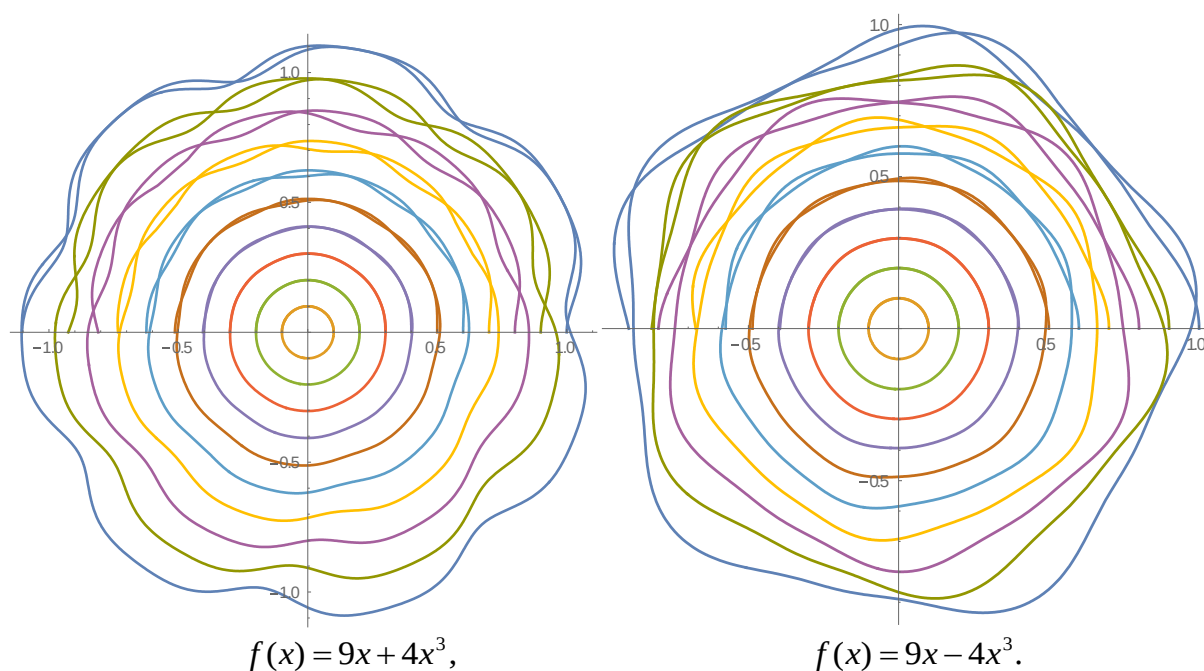
На рис.2 изображены соответствующие графики при $f(x) = 9x + 4x^3$, а на рис. 3 – графики при $f(x) = 9x - 4x^3$. В обоих случаях $f'(0) = 9$, а значит, $\omega = 3$. Поэтому период

$T(0) = 2\pi/3$. Принимая $a_0 = 0.1, 0.2, 0.3$, и т.д., получим зависимость периода $T(a)$.

Рисунок 2. — Фазовая траектория и график при $f(x) = 9x + 4x^3$.

Рисунок 3. — Фазовая траектория и график при $f(x) = 9x - 4x^3$.

На рис. 4 изображены фазовые траектории при различных начальных значениях $Z_0 = a$.

Рисунок 4. — Фазовые траектории при различных начальных значениях $Z_0 = a$.

Есть в математических пакетах решение дифференциальных уравнений для комплексной функции z действительного аргумента t . поэтому можно получать траекторию для $z(t)$, находя период $T(a)$, исходя от $T(0)$.

Я.Г. Пановко [2,3] предложил метод прямой линеаризации нелинейной функции $f(x)$ заменой линейной функции kx , которая наиболее близка в смысле среднего квадратического с весом $\varphi(x)$:

$$\int_0^a [f(x) - kx]^2 \varphi(x) dx - \min, \quad \text{Отсюда}$$
 получаем при $\varphi(x) := \omega^2 x + mx^3, \varphi(x) = x$

$$\omega^2(a) = \omega^2 + 0,4ma^3, \quad \text{а при } \varphi = 1$$

$$\omega^2(a) = \omega^2 + 0,5ma^3, \quad (7)$$

Весовая функция $\varphi(x) = x$: положительна и придаёт больший вес большим значениям x . В нашем случае $f(x) = 9x \pm 4x^3$ и поэтому $\omega^2(a) \approx 9 \pm 0,4 \cdot 4a^3$. Данные, снятые с правых графиков типа рис. 2 и 3, позволяют оценить точность формул (7), используя равенство $\omega = 2\pi/T$.

На рис.5 показаны графики $\omega(a)$ по формулам (7) и их «экспериментальные» значения, снятые с графиков рис. 2 и 3.

2 Период колебаний – функция амплитуды консервативной системы

Из уравнения (1) для нечётной функции $\varphi(x)$ получаем $T(a) = 4 \int_0^a \frac{dx}{\sqrt{2\Phi(a) - 2\Phi(x)}}$, где $\Phi(x)$ – первообразная функции $\varphi(x)$, $\Phi'(x) = \varphi(x)$. Интеграл является несобственным, т.к. подынтегральная функция обращается в бесконечность в правой точке интервала интегрирования $x = a$. Используя аддитивность интеграла, получим, выделив малый отрезок $(a - e, a)$. Для основной части отрезка $[0, a - e]$ интеграл можно вычислить по одной из формул численного интегрирования, а на конечной части отрезка $(a - e, a)$ использовать формулу интегрирования

$$\int_b^a \frac{dt}{\sqrt{c^2 - t^2}} = \arcsin \frac{a}{c} - \arcsin \frac{b}{c}, \quad \text{используя}$$

приближение функции $\Phi(x) \equiv \Phi_2(x) = \Phi(a) + \varphi(x)(x - a) + 0,5\varphi'(a)(x - a)^2$ её многочленом Тейлора в окрестности точки a . Последний равен

$$\frac{4}{\sqrt{\varphi'(a)}} \arccos(1 - e\varphi'(a)/\varphi(a)).$$

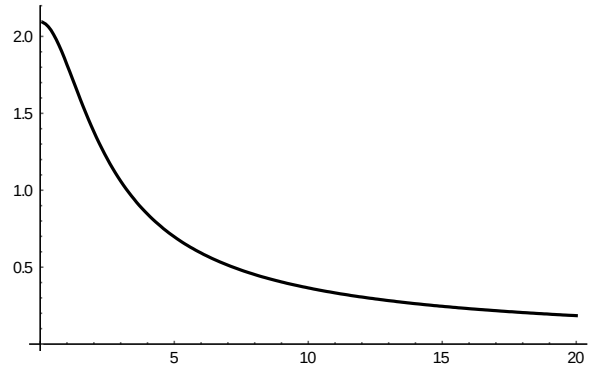


Рисунок 5а. — Зависимость периода для $f(x) = 9x + 4x^3$.

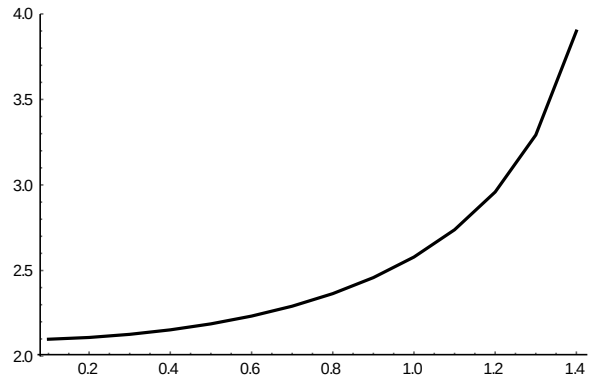


Рисунок 5б. — Зависимость периода для $f(x) = 9x - 4x^3$.

3 Более общее нелинейное автономное уравнение

$$\begin{aligned}\ddot{x} + f(x, \dot{x}) &= 0, f(0,0) = 0, \\ f'_x(0,0) &= \omega^2 > 0\end{aligned}\quad (8)$$

может быть аналогично сведено к уравнению первого порядка для комплексной функции $z(t)$:

$$\begin{aligned}\dot{x} &= -\omega y, \\ \dot{y} &= \omega x + (f(x - \omega y) - \omega^2 x) / \omega.\end{aligned}\quad (9)$$

В результате перехода к комплексной функции $z(t)$, получаем уравнение

$$\begin{aligned}\dot{z} &= i\omega z + [f(\operatorname{Re} z, -\omega \operatorname{Im} z) - \\ &- \omega^2 \operatorname{Re} z] / \omega.\end{aligned}\quad (10)$$

В качестве примера рассмотрим **уравнение Ван дер Поля**

$$\ddot{x} + x = e\dot{x}(1 - x^2), 0 < e \ll 1. \quad (11)$$

Используя формулу (10), получим уравнение

$$\dot{z} = iz + ie \operatorname{Im} z (1 - (\operatorname{Re} z)^2). \quad (12)$$

Интересно получить фазовые траектории и графики колебаний при разных начальных условиях и различных значениях «малого параметра».

Рассмотрим начальные значения $z(0) = 0.1; 5$ и параметра $e = 0.1; 1; 5$. Известно, что уравнение (11) имеет *предельный цикл*, для которого $z(0) = 2$, поэтому выбраны начальные точки *внутри* цикла и *вне* его. Предельный цикл устойчив орбитально.

Рис. 6 и 7а, 7б иллюстрируют сказанное.

4 Неавтономное уравнение с гармоническим возбуждением

$$\begin{aligned}\ddot{x} + f(x, \dot{x}) &= E \sin(pt), f(0,0) = 0, \\ f'_x(0,0) &= \omega^2 > 0\end{aligned}$$

аналогично сводится к уравнению первого порядка с комплексной функцией

$$\begin{aligned}\dot{z} &= i\omega z + [f(\operatorname{Re} z, -\omega \operatorname{Im} z) - \\ &- \omega^2 \operatorname{Re} z - E \sin pt] / \omega.\end{aligned}\quad (13)$$

Тут интересно рассмотреть резонансный и не резонансный режимы. В качестве функции $f(x, \dot{x})$ можно взять $9x \pm 4x^3 - \alpha \dot{x}$, где 2, 3, 5 – дорезонансный, резонансный и зарезонансный режимы. Тут проявится переходной процесс и установившийся режим. Длительность переходного процесса зависит от коэффициента сопротивления $\alpha > 0$.

Для линейной системы при отсутствии сопротивления получаем уравнение в комплексной форме

$$\begin{aligned}\dot{z} - i\omega z &= E / \omega \cdot e^{ipt} \Rightarrow \\ z &= C e^{i\omega t} + i \frac{E / \omega}{p - \omega} e^{ipt}\end{aligned}$$

Имеем при $\omega \neq p$ (нерезонансный режим).

Результаты решения уравнения (13) показаны на рис. 8-10 при $\alpha = 0.1$ для функции $9x - 4x^3 - \alpha \dot{x}$.

Для функции $9x + 4x^3 - \alpha \dot{x}$ результаты представлены на рис. 11.

Значение E равно 1.

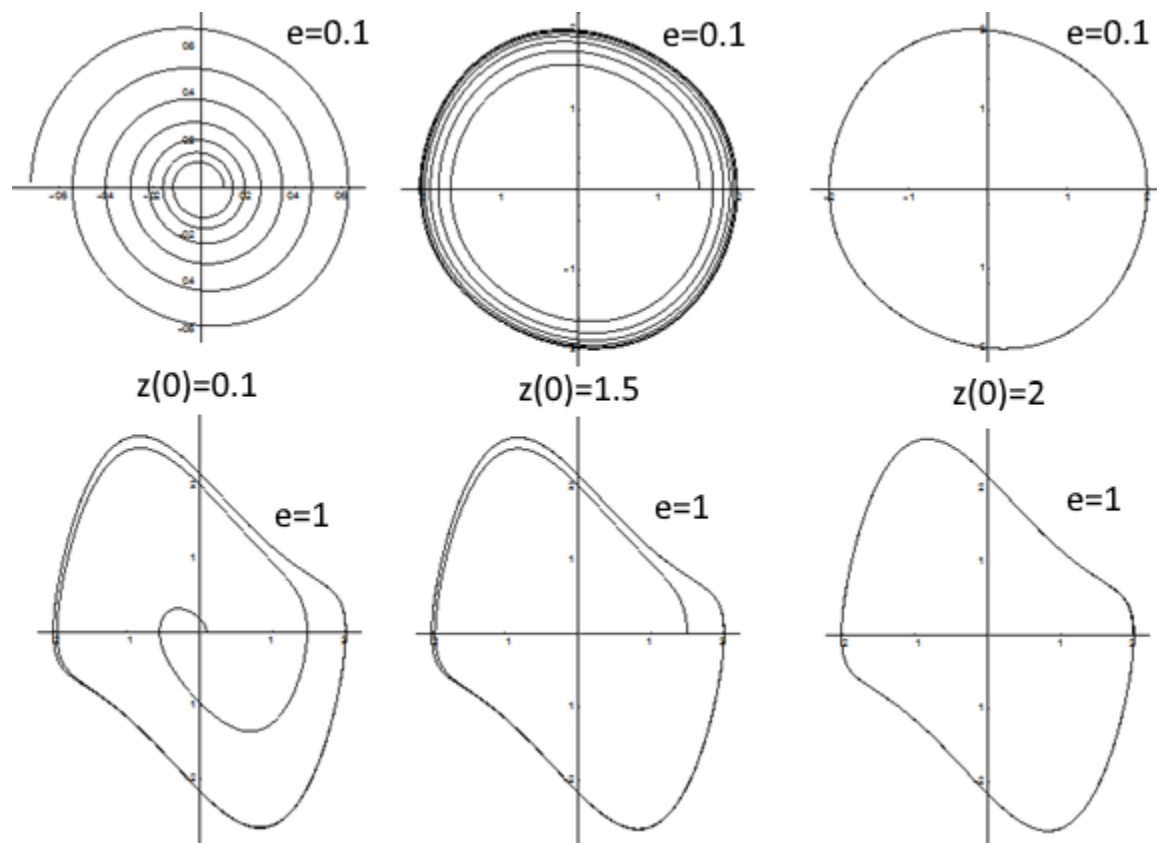
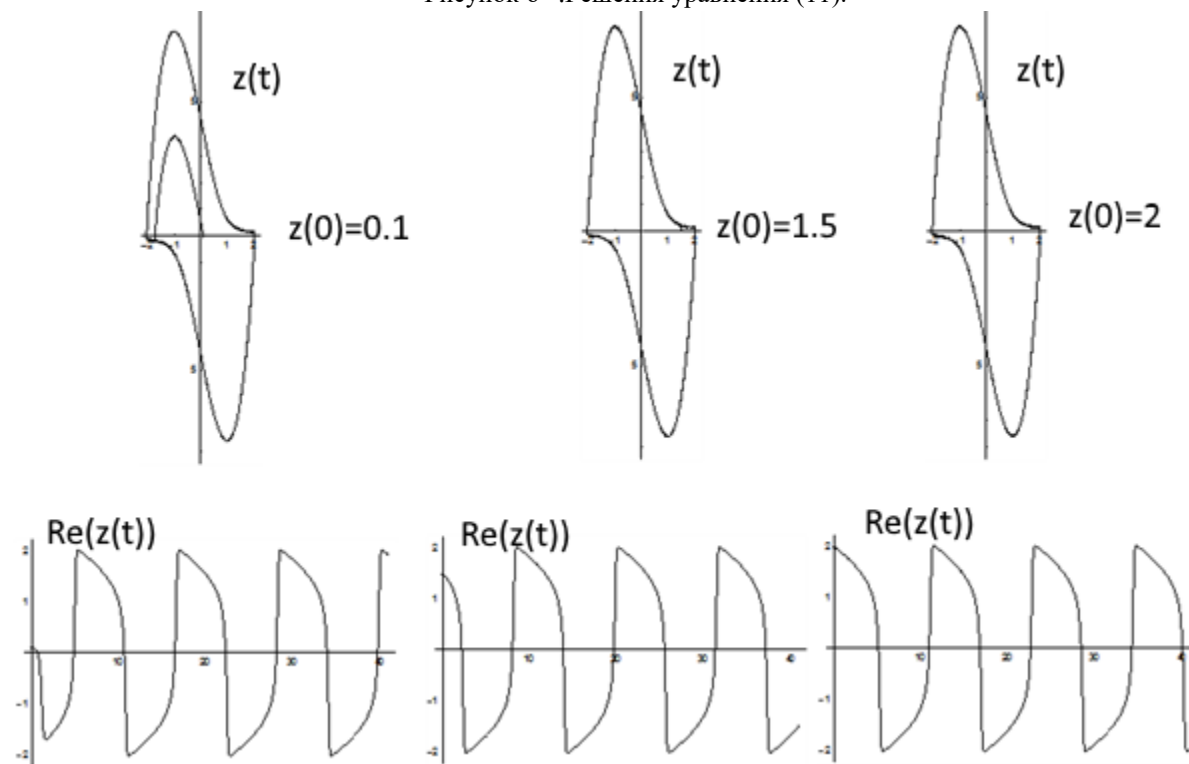


Рисунок 6—Решения уравнения (11).

Рисунок 7а—Решения уравнения (11) при $e=5$.

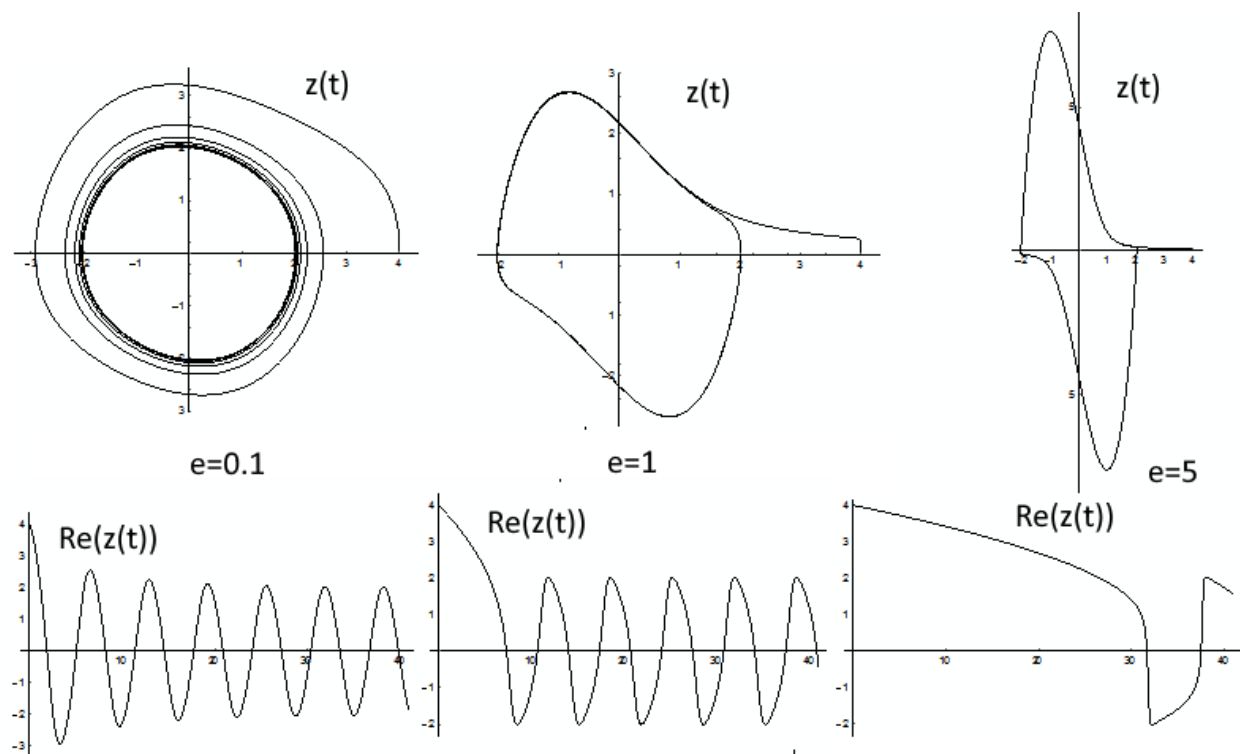
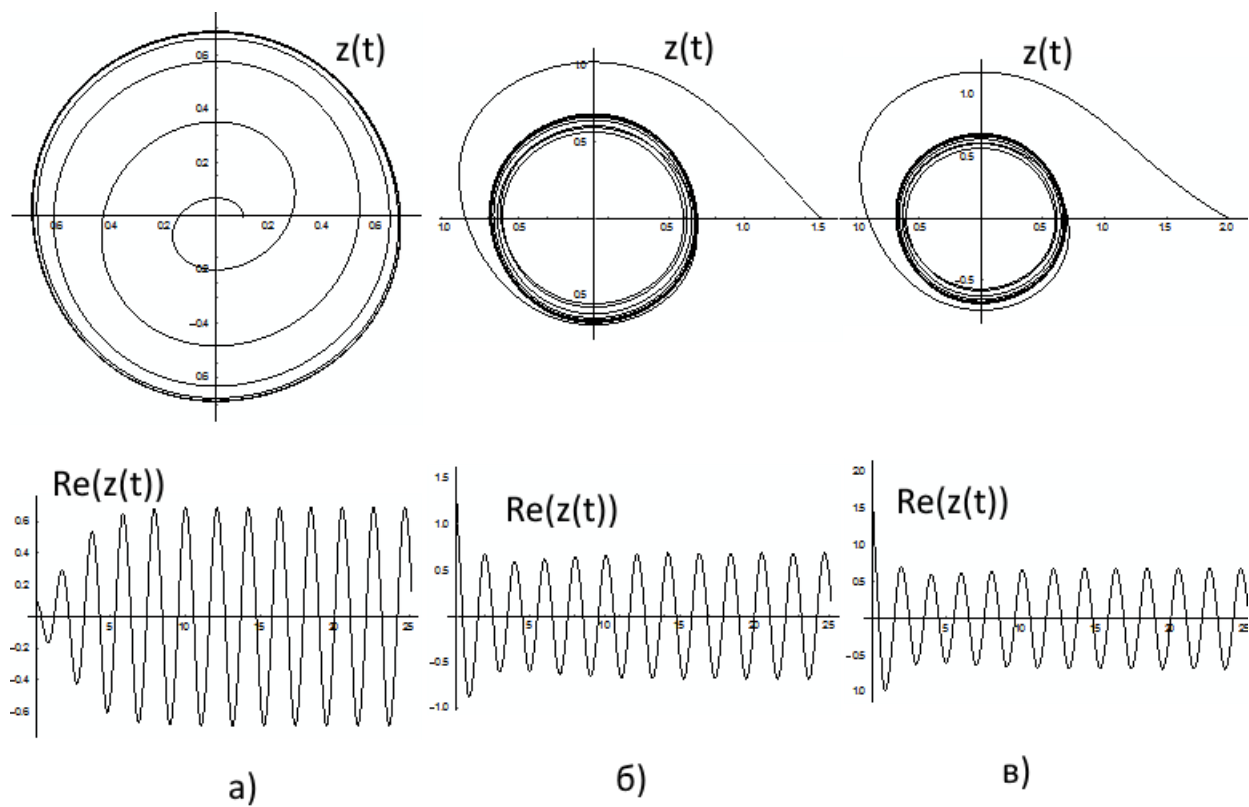
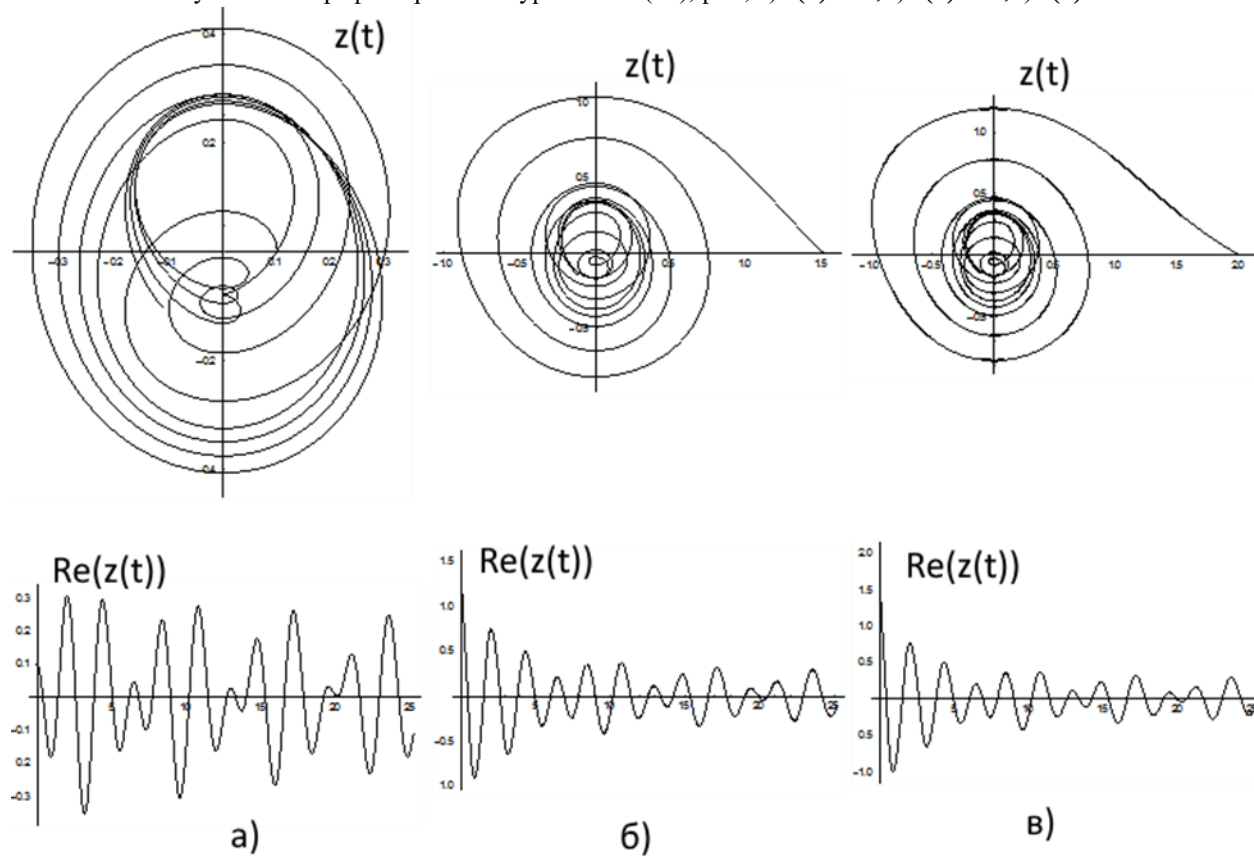
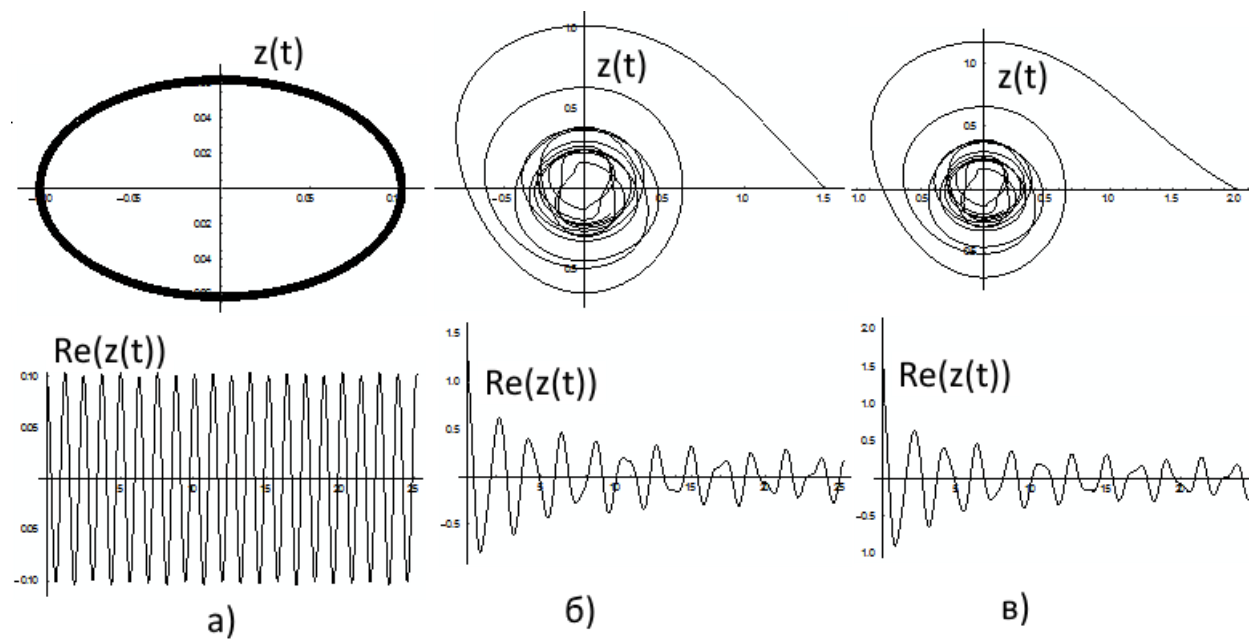
Рисунок 76. — Решения уравнения (11) при $z(0)=4$.

Рисунок 8. — Графики решения уравнения (13), $p=3$, а) $z(0)=0.1$; б) $z(0)=1.5$; в) $z(0)=2$.Рисунок 9. — Графики решения уравнения (13), $p=2$, а) $z(0)=0.1$; б) $z(0)=1.5$; в) $z(0)=2$.Рисунок 10. — Графики решения уравнения (13), $p=5$, а) $z(0)=0.1$; б) $z(0)=1.5$; в) $z(0)=2$.

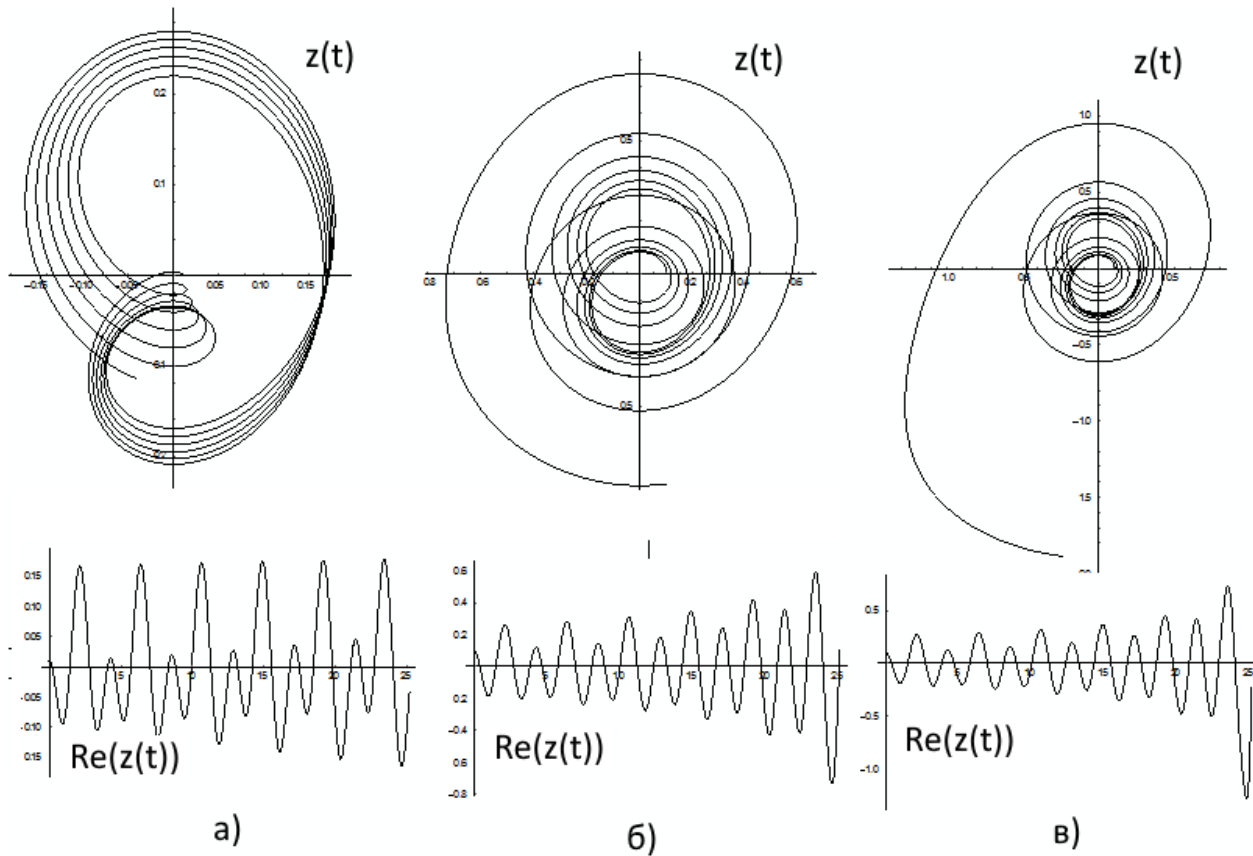


Рисунок 11. — Графики решения уравнения (13), $p=1.5$, а) $z(0)=0.01$; б) $z(0)=0.1$; в) $z(0)=0.105$.

Заключение

Показана возможность и целесообразность использования комплексной формы дифференциальных уравнений для анализа колебательных процессов с визуализацией результатов[4].

2. Переход к комплексным переменным позволяет снизить вдвое порядок соответствующих дифференциальных уравнений.

3. Использование современных математических пакетов типа Wolfram Mathematica упрощает процесс получения визуализации решений[5].

4. Желательно использование полученных результатов для анализа колебаний систем с несколькими степенями свободы.

Литература

1. Вибрации в технике. Справочник в 6 т. Т.1. Колебания линейных систем/ Под ред. В.В. Болотина. - М.: Машиностроение, 1978. – 352 с.
2. Пановко Я.Г. Основы прикладной теории упругих колебаний. – М.: Машиностроение, 1967. – 316 с.
3. Пановко Я.Г. Введение в теорию механических колебаний. – М.: Наука, 1991. – 256 с.
4. Дифференциальные уравнения. Электронный учебно-методический комплекс по дисциплине «Дифференциальные уравнения» для студентов математического факультета/ Н.А.Кучер, Д.А.Прокудин, О.В.Мальшенко. – Кемерово: Кемеровский госуниверситет.

5. Diran Basmadjian and Ramin Famood .
The Art of Modeling in Science and Engineering
with Mathematica, Second Edition, YiGU Hall/CRC
,2006

Filer Z.E., Andryukhin AI Computer investigation and visualization of vibrational dynamics of systems in complex form. We consider fluctuations through the use of an integrated form of the equation. linear parts of these equations are determined. Their subsequent integration is performed. Phase portraits and motion graphics built using Wolfram Mathematica package. The calculation results are shown

Keywords: vibrations, non-linear, complex equations

Филер З.Е., Андриюхин А.И. Компьютерное исследование и визуализация колебательной динамики систем в комплексной форме. Рассматриваются колебания с помощью использования комплексной формы уравнения. Выделяются линейные части. Выполняется их последующее интегрирование. Выполняется. Фазовые портреты и графики перемещений строятся с использованием пакета Вольфрам Матхематика . Результаты расчетов приведены

Ключевые слова: колебания, нелинейные, комплексные уравнения

Статья поступила в редакцию 12.12.2016
Рекомендована к публикации д-ром физ.-мат.. наук С.Н. Судаковым

УДК 004.588

Электронное обучение (e-Learning)

В.А. Черкасов, В.Н. Азаров

Национальный исследовательский университет Высшая школа экономики

Cherk-vlad@yandex.ru; Vazarov@hse.ru

Черкасов В.А., Азаров В.Н. Электронное обучение (e-Learning). Поставлены задачи по исследованию систем и технологий электронного обучения в России. Рассмотрена технология создания электронных курсов на основе зарубежного опыта. Представлено и рассмотрено ряд систем электронного обучения, а также средств по созданию и управлению учебным контентом. Обозначены минусы и плюсы электронного образования. Проведены различия между LMS и LCMS. Сделаны выводы и предложены решения проблемы развития информационно-технологических систем в России.

Введение

В настоящее время наиболее актуальным становится внедрение информационно - телекоммуникационных технологий в учебный процесс, поскольку для специальных категорий обучающихся, которые по объективным причинам не могут посещать образовательные учреждения, данный способ получения знаний представляется наиболее удобным и практичным. Система электронного обучения, ее еще называют «Electronic Learning» (e-learning), в зарубежных странах практикуется достаточно давно и сейчас она является преобладающей формой образования. В России данная форма обучения пока только набирает свою популярность.

Сложность данной технологии состоит том, что она требует привлечения значительных ресурсов, не только материальных, но и людских. Над созданием и реализацией данных систем трудится не один десяток программистов, например, такой как системы дистанционного тренинга Redclass, а также «Прометей». Сейчас в эпоху информационных технологий, электронное обучение является оптимальным решением для студентов, не только с точки зрения планирования своего времени, но и, с экономической, так как цена за электронные курсы существенно ниже. Опрос журнала Training Magazine показал, что компании, которые используют e-Learning для обучения персонала, экономят от 50 — 70% по сравнению с теми, которые придерживаются наиболее распространённой модели.[1]

Отталкиваясь от того, что сама технология создания систем электронного обучения не так проста, я решил исследовать данный вопрос на примере зарубежного опыта, так как на данный

момент, система электронного обучения достаточно широко используется в различных зарубежных школах, университетах и других образовательных учреждениях.

Постановка проблемы

С технической точки зрения для создания системы электронного образования требуются знания в области программирования, навыки по работе с программным обеспечением, опыт работы с платформами и знание принципов их совместимости. С точки зрения наполнения курсов информацией, требуются знания различных педагогов, квалифицированных специалистов обладающих высоким уровнем знаний в той или иной области. Основная проблема заключается в выборе подходящего программного обеспечения для систем электронного обучения, совмещение систем управления учебными курсами LMS и LCMS.

Актуальность темы объясняется тем, что в России сейчас происходит переход на электронную форму обучения как основную. Но, к сожалению, проблема заключается не только в техническом плане, но и в том, что отсутствуют единые интегрированные учебные электронные программы, а также электронно-методические пособия, но сейчас прикладываются все усилия для достижения обозначенной цели. Не смотря на то, что в огромное количество образовательных учреждений были подключены к единой локальной информационной сети и проведена работа по совершенствованию системы обучения, электронное образование России все же значительно уступает зарубежным информационно-технологическим системам.

Цель статьи - произвести различия между электронным и дистанционным обучением, проанализировать процесс внедрения

электронного обучения в России. Вывести плюсы и минусы электронного обучения при обзоре системы создания и управления учебным контентом.

Дистанционное и электронное обучение

Понятие электронного обучения в России закреплено Федеральным законом от 29.12.2012 N 273-ФЗ "Об образовании в Российской Федерации", который гласит: «Под электронным обучением понимается организация образовательной деятельности с применением содержащейся в базах данных и используемой при реализации образовательных программ информации и обеспечивающих ее обработку информационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу по линиям связи указанной информации, взаимодействие обучающихся и педагогических работников». [2]

Первоначально, в конце 20-х годов в России появилось дистанционное обучение, суть которого заключается во взаимодействии обучающихся и педагогических работников даже без применения информационно-телекоммуникационных сетей, например, посредством телерадиовещания, пересылки материалов по почте и другие. В основе дистанционного обучения лежит интенсивная самостоятельная работа обучаемого, который сам устанавливает свой график обучения. Основной плюс данной формы в том, что обучаемый в любое время прохождения обучения может проконсультироваться с преподавателем.

Electronic Learning предполагает, создание электронной информационно-образовательной среды, с помощью которой, обучающийся самостоятельно осваивает курсы. [3] В процессе электронного обучения используются интерактивные электронные средства доставки информации, преимущественно Интернет и корпоративные сети компаний, но не исключены и другие способы, как, например, CD-диски.

Из вышесказанного, можно определить, что целью электронного обучения является передача образовательных знаний независимым обучаемым через платформу онлайн-курсов. Учащийся является независимым, поскольку его никто не ограничивает по времени прохождения курса, а также никто не предоставляет дополнительных консультаций при возникновении вопросов. Курс состоит из интерактивной обучающей информации, такой

как мультимедийные презентации, графические технологии, расположенной и поддерживаемой на Web – сервере и, с помощью которого, обеспечивается доступ к содержимому курса. Основные преимущества Electronic Learning выделены на рисунке 1.

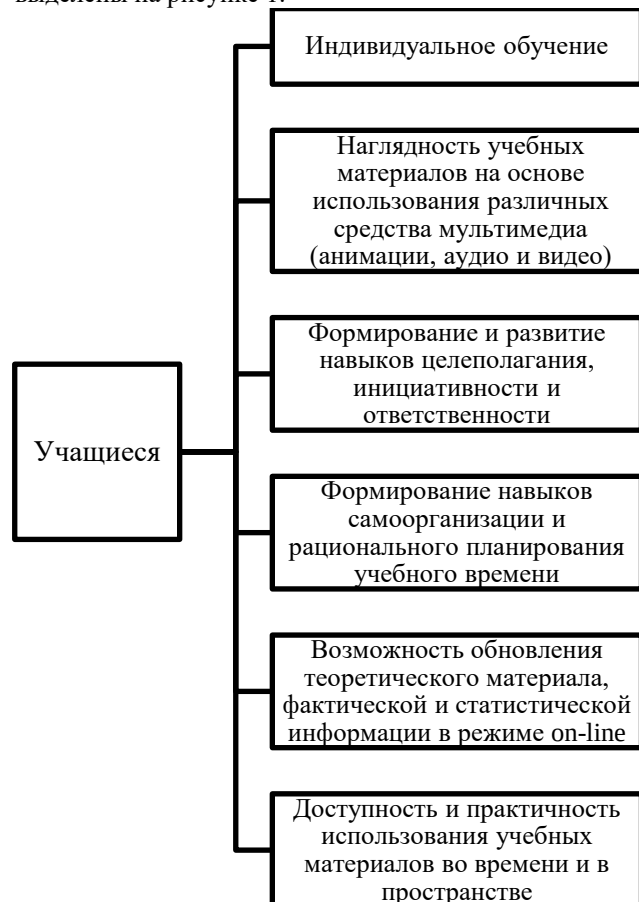


Рисунок 1 - Преимущества электронной формы образования для учащегося.

В 2004 году был проведен опрос компаний SkillSoft [4], в котором участвовали студенты различных вузов. По результатам опроса, были выявлены одни из наиболее важных плюсов электронного обучения: гибкость, экономия времени, простота возвращения к пройденному учебному материалу.

Проблема кроется в опыте использовании электронного обучения. Не все образовательные учреждения настроены на применение информационно - телекоммуникационных технологий в учебном процессе, поскольку это требует материальных ресурсов, но и затрагивает квалификацию преподавателей составляющих программы данных курсов. Такие выводы были сделаны на основе Информационного материала Министерства образования и науки РФ от 19 мая

2014 года [5] в преддверии парламентских слушаний.

Также можно сделать вывод о том, что обучаемый не находится под каким либо присмотром или контролем, в связи с чем невозможно обеспечить достаточную мотивацию для прохождения курса до конца. Хотя исследователь Джон Келлер излагал о том, что сначала нужно привлечь внимание обучаемого, затем нужно убедить его в важности и значимости обучения, после чего необходимо поддержать его уверенность в себе и, в конце концов, добиться удовлетворенности учащегося. Эффективность модели к обучению ARCS подкреплена результатами исследования: через год после внедрения этой технологии число участников, бросивших учебу, снизилось вдвое с 44% до 22%. [6]

Последний недостаток e-Learning заключается в отсутствии обратной связи между преподавателем и обучаемым, так как нет вербального общения, поэтому электронное обучение имеет определенные ограничения в применении. Например, оно не подходит для развития навыков работы в команде, уверенности и коммуникабельности

Структура системы e-Learning

Два стандартных модуля полномасштабной системы электронного обучения:

1. Системы управления обучением (LMS - learning management system)
2. Системы управления учебным контентом (LCMS - Learning Content Management System)

Программное обеспечение должно быть правильно подобрано, так как оно выполняет организацию и поддержку учетных записей пользователей, предоставляет возможность создавать и настраивать учебный контент. Оно может быть представлено как простыми статическими HTML страницами, так и сложными системами управления обучением (LMS) и учебным контентом (LCMS), использующимся в корпоративных компьютерных сетях.

Успешное внедрение электронного обучения основывается на правильном выборе программного обеспечения, соответствующего конкретным требованиям, целям и задачам, которые предъявляются к нему организацией [7]. И в связи с этим к средствам организации электронного обучения применяются некоторые требования. Основными из них являются:

- **Функциональность.** Это достаточно важное требование, поскольку от этого зависит прагматичность представленных курсов, которая выявляется в наличии в системе набора функций различного уровня, таких как форумы, чаты, анализа активности обучаемых, управления курсами и обучаемыми и другие;
- **Удобство администрирования,** простота обновления учебного контента и стабильность системы. Это в первую очередь влияет на отношение пользователей к данной системе, а также эффективность ее использования;
- **Стоимость,** которая складывается из цены самой системы, а также из затрат на ее внедрение, разработку курсов и сопровождение, наличие или отсутствие ограничений по количеству лицензий на обучаемых;
- **Удобство использования** для обучаемых, поскольку данная система должна быть интуитивно понятной;
- **Мультимедийность,** означает наличие аудио, видео, gif- и flash-анимации, 3D-графики различных файловых форматов, а не только текстовых, гипертекстовых и графических файлов.

Далее на рисунке 2 показаны основные два типа сегмента рынка электронного образования.



Рисунок 2 - Сегмент рынка электронного образования.

Системы управления обучением (LMS)

Данные системы были специально созданы для того, чтобы отследить успеваемость обучаемых. В качестве примера таких систем можно привести два вида:

- корпоративные (Docent, Saba, Click2learn's Aspen);
- академические (Blackboard, e-College, WebCT).

LMS обеспечивает доступ пользователям к содержанию учебных курсов, а также она позволяет разработчику или администратору обучения осуществлять оперативный контроль над самим процессом обучения всех обучающихся, это возможно с помощью разграничения прав доступа к системе. Также LMS можно определить как платформу для развертывания e-Learning, которая в ряде случаев может использоваться и для администрирования традиционного учебного процесса.

Обучаемые с помощью индивидуального имени пользователя и пароля, которые они получили при прохождении регистрации, имеют доступ к учебному ресурсу, а также к статистике своего обучения. LMS регистрирует: имя пользователя, пройденные им курсы, время начала и конца изучения каждого курса, полученные баллы в результате предварительного тестирования и тестирования после прохождения обучения. Другими словами, пользователь получает от LMS возможность доступа к учебному portalу, который является отправной точкой для доставки всего учебного контента, выбора подходящих учебных курсов на основе предварительного и промежуточных тестирований, использования дополнительных материалов с помощью предназначенных для этого ссылок. Администратор нужен для контроля над доступом к системе, также он отвечает за составление отчетности по всей системе учебного процесса, на основе которой, в дальнейшем, возможно, сделать выводы об эффективности инвестиций в обучение.

Безусловно, для решения вопроса управления учебным процессом, LMS должна иметь возможность проигрывать разные готовые курсы от различных поставщиков. Для этого разрабатываются специальные стандарты интероперабельности. Достаточно известный и часто применяемый в данной области это стандарт SCORM (Sharable Content Object Reference Model) созданный компанией ADL (Advanced Distributed Learning), а также IEEE (Institute of Electrical and Electronics Engineers)

каждый из которых поддерживается, развивается и имеет всеобщее признание. Поддержание одного и того же стандарта, нужно для полноценной работы всех модулей системы электронного обучения.

В совокупности, система управления обучением LMS предоставляет следующие возможности: поддержки смешанного обучения, интеграции с HR, администрирования с помощью различных инструментов, интеграции контента, тестирования, управления знаниями.

Системы управления учебным контентом (LCMS)

Данные системы имеют своей задачей управление содержанием учебных курсов. Они разработаны для персонала трудящихся над созданием и компоновке курсов, а именно это: разработчики контента, специалисты по методологическому "наполнению" курсов и руководители проектов. Основой является концепция представления содержания обучения как совокупности многократно используемых учебных объектов со своей целевой аудиторией и определенным контекстом использования.

На данный момент, существуют различные варианты возможностей LCMS, исходя из этого выделяют четыре основных компонента данных систем.

Во-первых, это репозиторий учебных объектов, его определяют как основную (центральную) базу, которая хранит и управляет учебным контентом. Итоговый продукт может быть доступен через Web, CD-ROM, или в бумажном виде. Каждый объект, может быть использован несколько раз и с различными целями, это зависит от предъявляемых требований.

Во-вторых, необходимо программное обеспечение автоматизированного аутсорсинга, которое используется при создании неоднократно используемых учебных объектов, в последствии, которые потом будут доступны в репозитории. Эксперты, создатели медиа-продукции [8] а также иные авторы используют шаблоны и архивные образцы, которые содержат основные принципы дизайна, при разработке курсов, при этом они применяют имеющиеся объекты из репозитория, а также создают новые объекты, или используют комбинацию данных объектов.

В-третьих, это интерфейс проигрывания контента. Он создан для того, чтобы отображать материалы, которые запрашивает пользователь, например для предварительного тестирования. Этот компонент также обеспечивает трекинг

результатов, ссылки на источники информации, а также обратную связь для пользователей.

В-четвертых, средства администрирования как приложение, используемое для управления учетными записями, мониторинг результатов, запуск представленных курсов в каталоге, а также другие административные функции. Здесь, можно сказать о том, что данная информация может быть передана в LMS, так как она предназначена для администрирования системы в целом.

Минус LCMS систем состоит в том, что для проектирования курсов нужен большой опыт работы в данной системе, так как включая или убирая какие либо элементы из системы можно нарушить её целостность.

На данный момент, можно привести несколько примеров LCMS систем: Claro, Cornerstone OnDemand, Kenexa LCMS, SumTotal LCMS и другие.

Сравнение систем LMS и LCMS

Обе системы предназначены для управления учебный контентом. Также с учетом того, что обе системы придерживаются стандартов XML (eXtensible Markup Language — расширяемый язык разметки), возможна передача учебного контента между системами. Основным различием, помимо упомянутых на таблице 1, является то, что с помощью LMS можно управлять и настраивать несколько курсов, так как в LCMS данная функция недоступна. Но, зато в последней системе, возможно управление, а также подбор курсов на более глубоком уровне, учитывая потребности пользователей.

Таблица 1 - Сходства и различия между системами LMS и LCMS.

	LMS	LCMS
Для кого полезна?	Для всех пользователей	Для разработчиков контента; учащихся, которых интересует персонализированный контент
Предоставление возможности по управлению	Учебным процессом	Учебным контентом
Управление электронным обучением	Да, управляет каталогом курсов, расписани	Да, разрабатывает учебные объекты, интерфейс,

	ем, системой регистрации и студентов, результатов ми	методы оценивания и совместной работы
Мониторинг результатов	Да	Да
Поддержка управления профиля учащегося	Да	Нет
Инструменты создания тестов и их администрирование	Да	Да
Поддержка адаптивного обучения на основе интересов учащихся	Нет	Да
Поддержка создания контента	Нет	Да
Имеет все рабочие инструменты для создания и управления контентом	Нет	Да
Хранение данных	Хранит данные о курсах и студентах	Хранит контент в виде учебных объектов в репозитории

В заключение, хотелось бы сделать небольшой вывод по поставленной проблеме в начале моей статьи. Созданная система должна быть удобна и понятна пользователю. Если программа будет «громоздкой» и непонятной для учащегося, он просто на просто, не будет заинтересован в ней и предпочтёт что-то более практичное и удобное для него. В любой время, в любом месте и с любого устройства, то есть доступ должен быть не только со стационарного компьютера, но и со смартфона. Пользователи должны быть сконцентрированы на процессе обучения, а никак не на самой работе интерфейса. В настоящее время, существуют компании, которые занимаются разработкой программного обеспечения, которые совместимы с LMS и LCMS, так как данные системы созданы именно

для разработки и управления учебными курсами. Инструменты для продукта электронного обучения должны быть тщательно подобраны, для удовлетворения потребностей учеников и преподавателей.

Выводы

Исходя из всего вышесказанного, можно сделать несколько обоснованных выводов. Электронное обучение – это форма образовательного процесса, внедрение которой представляется возможной с помощью различных ИТ-технологий. Электронное образование развивается в России сравнительно недавно и, в связи с этим, пока не существует четко выработанной системы электронного обучения, которая могла быть основной в образовательных учреждениях или организациях.

Electronic Learning - это наиболее удобная форма обучения для пользователей, так как ее преимущества заключаются в гибкости, практичности, низкой стоимости, функциональности, наглядности и простоте. Но, также, данная технология имеет и минусы, такие как отсутствие вербального контакта преподавателя и обучающегося, отсутствие мотивации для прохождения курса до самого конца. К сожалению, отсутствие приспособленности у образовательных учреждений и застоявшаяся система дистанционного обучения тоже являются отрицательными факторами.

Для создания системы электронного обучения требуется несколько компонентов для разработки учебного контента, а также его управления, администрирования с целью сделать его практичным и доступным для различных пользователей. А конкретно, системы управления обучением (LMS - learning management system), учебного контента (электронных курсов). Также нужно упомянуть о разделении рынка электронного обучения на два основных сегмента, это компании разработчики программного обеспечения и компании по разработке и составлению электронных курсов. Взаимодействуя между собой, они создают единую систему электронного обучения для пользователей. Но проблема кроется в неприспособленности к данной форме обучения, что естественно сказывается на рынке предлагаемых услуг.

На данный момент электронное образование России все же значительно уступает зарубежным информационно-технологическим системам, данный вывод сделан на основании статистик и

исследований, представленных мною в данной статье. Здесь также стоит добавить, что отсутствие экономических ресурса для развития данной области, но и квалифицированных специалистов разрабатывающих данные курсы существенно малы. Решение данной проблемы состоит в создании интегрированных электронных курсов на основе зарубежного опыта, заимствование и использования подобных технологий, разработка реформы электронного образования в России, а также привлечение инвестиций в данную область.

Литература

1. Опрос журнала Training Magazine. [Электронный ресурс]. – Режим доступа: <https://trainingmag.com/>
2. Российская Федерация. Законы. Об образовании в Российской Федерации [Текст] : федер. закон : [принят Гос. Думой 21 декабря 2012 г. : одобр. Советом Федерации 26 декабря 2012 г.]. [Электронный ресурс].: http://www.consultant.ru/document/cons_d oc_LAW_140174/
3. Елена Тихомирова. Живое обучение: Что такое e-learning и как заставить его работать. М: Альпина Паблишер. 2013, 238 с.
4. Опрос компании SkillSoft. [Электронный ресурс]. – Режим доступа: <http://www.skillssoft.com/>
5. Государственная дума РФ. Комитет по образованию. Информационный материал Министерства образования и науки РФ. М, 2014 – 7-8 с.
6. Джон Келлер. Model of Motivation and Design. США, 1987. 8 с.
7. Комелина Е.В., Гусакова Т.М. Использование технологий web2.0 в учебном процессе вуза // Преподавание Информационных Технологий в России: Открытая всероссийская конференция [Электронный ресурс]. – Режим доступа: http://www.iteducation.ru/2009/reports/Ko melina_Gusakova.htm
8. Management Systems used in Education // Informatica Economică. 2012. Vol. 16(1). PP. 123-131.
9. Лоскутникова В.М. Антропологические аспекты информатизации образования // Открытое и дистанционное образование. Выпуск 1(9), 2003. С. 66 – 71.

10. Майкл Аллен. E-Learning: Как сделать электронное обучение понятным, качественным и доступным. М: Альпина Паблишер. 2014, 200 с.

11. Владислав Бюрюлев. Статья: Электронное образование. М, 2015. [Электронный ресурс]. – Режим доступа: <http://ra-kurs.spb.ru/2/0/2/1/?id=83>

Черкасов В.А., Азаров В.Н. Электронное обучение (e-Learning). *Поставлены задачи по исследованию систем и технологий электронного обучения в России. Рассмотрена технология создания электронных курсов на основе зарубежного опыта. Представлено и рассмотрено ряд систем электронного обучения, а также средств по созданию и управлению учебным контентом. Обозначены минусы и плюсы электронного образования. Проведены различия между LMS и LCMS. Сделаны выводы и предложены решения проблемы развития информационно-технологическим систем в России.*

Ключевые слова: электронное образование, программное обеспечение, учебный контент, системы управления обучением, средства для разработки электронных курсов.

Cherkasov. V.A., Azarov V.N. Electronic education e-learning. *Are set tasks on a research of systems and technologies of electronic training in Russia. The technology of creation of electronic rates on the basis of foreign experience is considered. It is provided and considered a row of systems of electronic training, and also means on creation and management of educational content. Minuses and pluses of electronic education are designated. Distinctions between LMS and LCMS are carried out. Conclusions are drawn and problem resolutions of development information and technological systems in Russia are proposed.*

Keywords: *electronic education, the software, educational content, learning management systems, means for development of electronic courses.*

*Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко*

УДК 004.822

Автоматическая генерация структуры Wiki-систем при помощи онтологий

В.К. Шестаков

Институт систем информатики имени А.П. Ершова СО РАН
shestakov@iis.nsk.su

Шестаков В.К. Автоматическая генерация структуры Wiki-систем при помощи онтологий.
В статье описываются инструментальные средства, поддерживающие технологию автоматической генерации структуры систем на основе Wiki-технологии при помощи онтологий предметных областей. Технология позволяет получать Wiki-системы с хорошей структурой и согласованной системой понятий.

Ключевые слова: онтология, Wiki, семантические технологии, генерация структуры информационных систем.

Введение

Для удовлетворения все возрастающих информационных потребностей пользователей разработаны разнообразные средства построения информационных систем. Одним из таких удобных и простых в использовании средств сбора и хранения информации являются Wiki-системы [1]. Они позволяют работать не только с текстовым, но и с мультимедийным контентом, имеют удобный и интуитивно понятный интерфейс, просты в освоении. Однако их большим недостатком является то, что они позволяют отслеживать в создаваемых информационных системах только структурную целостность ссылок, не обеспечивая при этом логической непротиворечивости и семантической согласованности [2] используемых в них понятий и отношений. Общая идея предлагаемого подхода состоит в создании инструментария, который бы обеспечивал автоматическую генерацию структуры Wiki-систем с согласованной системой понятий (семантически согласованных Wiki). Wiki с такими свойствами можно получить, если генерировать ее на основе логически непротиворечивой онтологии, описывающей предметную область будущей системы.

Обзор средств создания информационных Wiki-систем, использующих семантические технологии

На сегодняшний день существует достаточно много проектов создания информационных Wiki-систем, так или иначе связанных с семантическими технологиями. Хотя эти проекты нацелены на решение различных задач, общим для них является использование онтологий на тех или иных этапах построения или

функционирования информационных систем. Так как на данный момент существует большое количество средств создания Wiki-систем, то в этом обзоре мы сосредоточим наше внимание только на тех из них, которые в той или иной мере используют семантические технологии.

Сначала рассмотрим проект из Санкт-Петербурга [3], в котором в одном из первых начали применяться онтологии при построении таких информационных систем, как корпоративные Wiki-порталы. Исходя из того, что онтологии позволяют создавать более концептуально-ясные и целостные модели предметных областей, разработчиками было решено привязать контент Wiki-системы к онтологии, чтобы избежать проблем с плохой структурой портала, т.к. она в этом случае будет такого же качества, как и определяющая ее онтология. В рамках этого проекта был разработан прототип инструментария OntolingWiki, который содержит внутри себя Wiki-движок и для каждого концепта онтологии автоматически создает Wiki-страницу, в которую эксперт вносит информацию об описываемом понятии. При этом конечные пользователи видят онтологию, представленную в виде графа, могут осуществлять навигацию по ее понятиям и видеть содержимое Wiki-страниц, соответствующих выбранному понятию.

Недостатком OntolingWiki является то, что онтология, лежащая в основе созданной с помощью нее Wiki-системы, не может быть изменена или дополнена средствами самой Wiki-системы.

Наиболее распространенным инструментальным средством для создания Wiki-систем является популярный Wiki-движок MediaWiki [4]. Именно для этого универсального движка больше всего строится расширений, позволяющих использовать средства семантических технологий. Одной из первых была создана надстройка над MediaWiki, получившая название Semantic MediaWiki [5]. Это

расширение допускает возможность импорта онтологии [6] и позволяет использовать в Wiki-системе структуры и связи, определенные в ней. Однако предоставляемый Semantic MediaWiki модуль импорта онтологий на данный момент находится в стадии бета-версии и не включен в состав новых версий данного расширения. В связи с этим реализацию импорта онтологий пользователи Semantic MediaWiki должны выполнять самостоятельно.

Другим расширением MediaWiki является BOWiki [7] — основанный на онтологии семантический Wiki-движок. Первоначально он базировался на Semantic MediaWiki, а потом был реализован как «чистое» расширение MediaWiki. Достоинством BOWiki является то, что он позволяет пользователям рассматривать сущности, описанные Wiki-страницами, в качестве экземпляров онтологических категорий, определять новые отношения в Wiki-контенте, связывать Wiki-страницы, в том числе *n*-арными семантическими отношениями, запрашивать Wiki-страницы, удовлетворяющие определенным условиям, допускает импорт еще нескольких био-онтологий для ограниченного использования и обеспечивает экспорт Wiki-контента в OWL-представление.

BOWiki разрабатывался как средство для совместного создания и интеграции знаний в области био-информатики, поэтому его функционирование основано на онтологии высокого уровня General Formal Ontology (GFO) [8] в OWL-версии. В своей работе он использует также две онтологии, основанные на GFO, — базовую биомедицинскую онтологию (GFO-Bio) и онтологию функций (Ontology of Functions или OF). Как было сказано выше, возможность использования других (внешних) онтологий существенно ограничена, поэтому BOWiki не может рассматриваться в качестве инструмента для построения Wiki-систем произвольного вида.

Также для MediaWiki существуют расширения, работающие совместно с Semantic MediaWiki, и добавляющие дополнительную функциональность по импорту семантической информации. Одним из таких является RDFIO [9]. Оно позволяет импортировать произвольные RDF-триплеты [10]. Но это расширение до сих пор имеет статус бета-версии и еще не готово для массового использования. Еще одним подобным расширением является LinkedWiki [11]. Оно применяется для получения данных с внешнего сервиса и отображения этих данных в Wiki-системе. Это расширение не может изменить структуру существующей системы, а только дополняет ее, поэтому имеет довольно ограниченную область применения.

Основные части Wiki-систем и инструменты для работы с ними

В основе любой Wiki-системы лежит так называемый Wiki-движок — комплекс программных средств для преобразования Wiki-разметки в код, предназначенный для отображения в браузере. Одним из самых распространенных движков является MediaWiki (на его основе работает широко известная Википедия). Для него существуют специальные дополнения, называемые расширениями, позволяющие получить определенную функциональность. Так, расширение Semantic MediaWiki дает возможность добавлять семантическую информацию за счет расширения разметки, а также предлагает средства для работы с этой информацией.

Для проведения работ технического характера в Wiki-системах используют ботов — специальные программы для выполнения заданного набора операций. Они являются клиентскими приложениями, поэтому не требуют внесения изменений на стороне сервера (т.е. для их работы не нужно модифицировать движок или ставить какое-либо расширение). Например, в Википедии ботов используют для таких задач, как переименование категорий и статей, расстановка интервики ссылок (ссылок на родственные проекты), исправление ссылок, удаление спама и т.п. Для реализации ботов используются различные языки программирования, а также существуют разные библиотеки для облегчения их написания. Одной из наиболее развитых библиотек является Python WikipediaBot Framework [12]. Она использует MediaWiki API (специальный интерфейс прикладного программирования) для взаимодействия с MediaWiki-системой: авторизации, получения данных и внесения изменений.

Автоматическая генерация структуры Wiki-системы при помощи онтологий: описание работы системы

Предлагаемая в рамках данной работы инструментальная система позволяет на основе ранее разработанной онтологии автоматически сгенерировать Wiki-систему, структура и содержание которой будет определяться этой онтологией. В дальнейшем построенная Wiki-система может расширяться как обычная Wiki-система с использованием традиционных средств Wiki-технологии.

Общая схема функционирования системы, реализованной в рамках предлагаемого подхода, объединяющего Wiki-технологии и онтологии, выглядит следующим образом (см. рис. 1). На первом этапе квалифицированный эксперт описывает онтологию предметной области. На втором при помощи разработанного

автором инструмента (Onto2Wiki) по ней строится интерфейс на основе Wiki-технологии, состоящий из двух частей: конструкторского и пользовательского. При помощи первого эксперты предметной области могут вносить данные в систему, которые тут же становятся

На первом шаге онтология, полученная в готовом виде или разработанная в Protégé [13] или каком-либо другом редакторе онтологий, сохраняется или конвертируется в OWL-формат [14]. После этого файл со спецификацией онтологии подается на вход специально

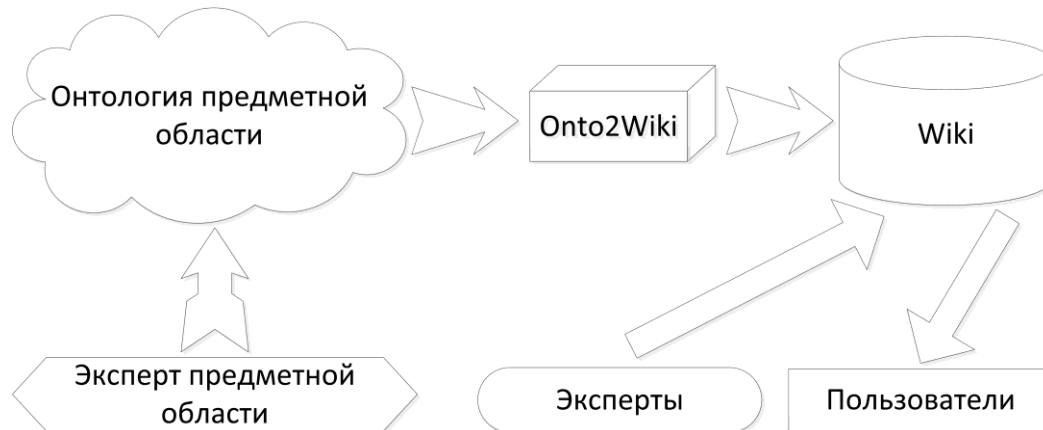
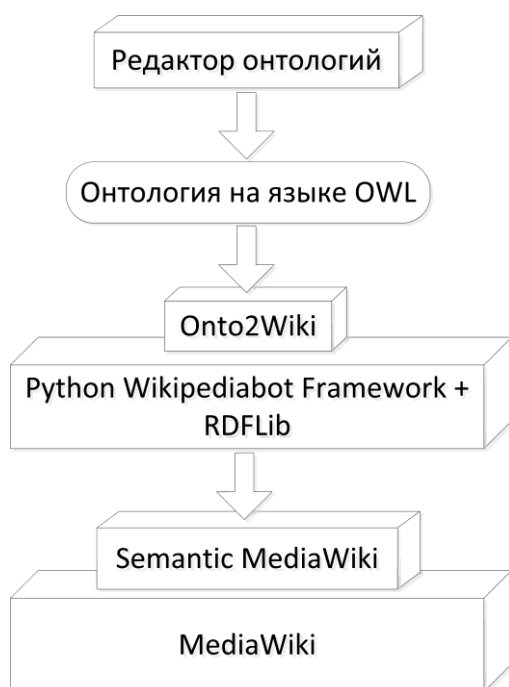


Рисунок 1 – Общая схема функционирования системы

доступны пользователям посредством второго.

Подробная схема функционирования инструментальной системы представлена на рис. 2. Согласно ей разработка Wiki-системы в рассматриваемой инструментальной системе включает следующие этапы.



5 Рисунок 2 – Подробная схема функционирования системы

разработанному в рамках данного проекта программному модулю Onto2Wiki, который разбирает его с использованием библиотеки RDFLib [15], а затем, при помощи среды Python WikipediaBot Framework, создает каркас информационной системы на основе пустого Wiki-сайта, работающего на базе MediaWiki с расширением Semantic MediaWiki. При этом в Wiki-систему добавляются нужные страницы, для которых указываются соответствующие атрибуты, расставляются категории и прописываются нужные связи. После выполнения этих действий Wiki-система готова к использованию.

В таблице 1 представлено соответствие между конструкциями языка OWL и семантической Wiki, которое используется при отображении онтологии в структуры и содержание Wiki-системы.

Таблица 1. Соответствие конструкций языка OWL и Semantic MediaWiki

Конструкция OWL	Semantic MediaWiki
Класс	Категория
Подкласс	Подкатегория
Экземпляр	Страница
Атрибут	Атрибут
Отношение	Типизированная ссылка

В соответствии с этой таблицей для каждого класса онтологии будет создана своя категория, которая будет проставлена на каждой подкатегории, соответствующей подклассу данного класса, и на каждой странице,

соответствующей экземпляру данного класса. Также все атрибуты экземпляра онтологии будут указаны в качестве атрибутов на странице, соответствующей этому экземпляру. При этом все отношения онтологии отобразятся в типизированные ссылки [16] между соответствующими страницами и категориями.

Также при отображении онтологии ее типы данных преобразуются в типы данных Semantic MediaWiki. Пример соответствия некоторых типов данных представлен в таблице 2.

Таблица 2. Пример соответствия типов данных при отображении

Тип OWL	Тип Semantic MediaWiki
string	Строка
double	Число
boolean	Булево
dateTime	Дата

Стоит заметить, что созданную с помощью данного инструмента систему можно дорабатывать в соответствии с возникающими потребностями, взаимодействуя с ней как с обычной Wiki-системой. Т.е. можно создавать

система получается полностью независимой от той онтологии, на основе которой была построена.

Демонстрация работы системы

Рассмотрим работу данной инструментальной системы на примере построения учебной информационной системы «Технология создания экспертных систем». Онтология для этой системы была построена в уже упоминавшемся редакторе онтологий Protégé. Ее визуализация представлена на рис. 3.

Вот фрагмент этой онтологии в формате Turtle:

```
ex:Как_построить_свою_ЭС      rdf:type
ex:Литература по ЭС ,
```

```
owl:NamedIndividual ;
    ex:название_книги
"Как построить свою экспертную
систему" ;
    ex:автор_книги
"Нейлор К." ;
    ex:упоминается
ex:MYCIN ,
```

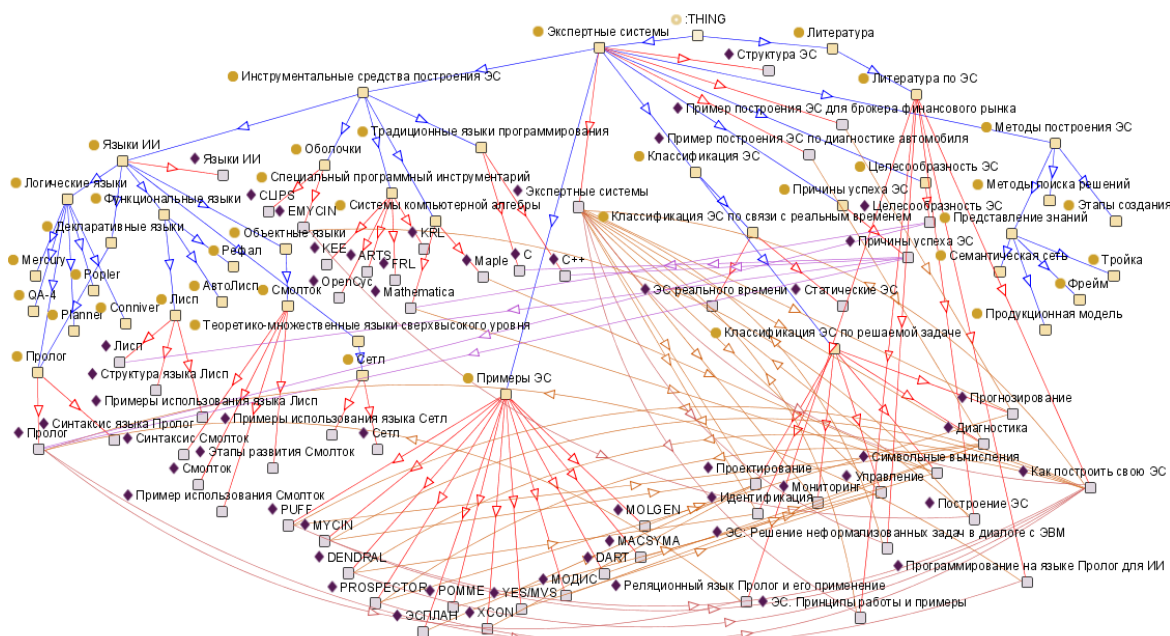


Рисунок 3 – Онтология «Технология создания экспертных систем».

новые страницы, прописывать новые связи, указывать новые категории и т.д.

Таким образом, особенностью данного подхода является то, что полученная Wiki-

ex:DENDRAL ,

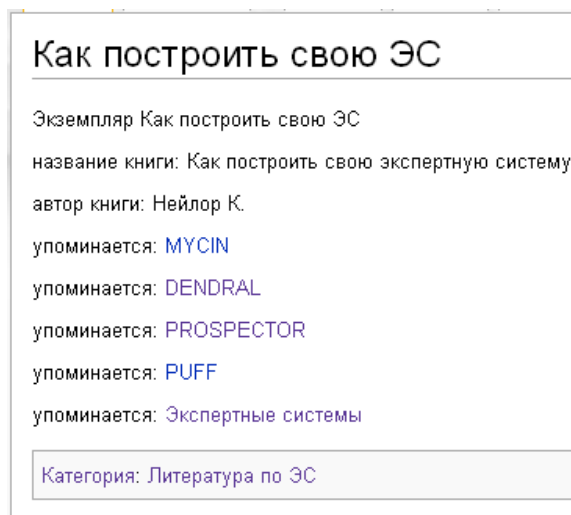
ex:PROSPECTOR ,

ex:PUFF ,

ex:Экспертные_системы .

Подаем эту онтологию на вход системе и после окончания ее работы получаем заполненную Wiki-систему.

На рис. 4 представлен пример одной из страниц полученной системы. На нем видны атрибуты и типизированные ссылки, находящиеся на странице.



6

7 Рисунок 4 – Одна из страниц

А вот так выглядит Wiki-разметка данной страницы:

```

Экземпляр Как построить свою ЭС

название книги: [[название
книги::Как построить свою экспертную
систему]]

автор книги: [[автор книги::Нейлор
К.]]

упоминается: [[упоминается::MYCIN]]

упоминается:
[[упоминается::DENDRAL]]

упоминается:
[[упоминается::PROSPECTOR]]

упоминается: [[упоминается::PUFF]]

упоминается:
[[упоминается::Экспертные системы]]

[[Категория:Литература по ЭС]]

```

Заключение

В данной работе рассмотрен способ автоматической генерации структуры систем на основе Wiki-технологии при помощи онтологий предметных областей. В рамках этого подхода предложен метод автоматической генерации структуры Wiki-систем при помощи онтологий, разработан прототип инструментальной системы, реализующий метод в объеме, указанном в четвертом разделе.

Применение метода автоматической генерации при помощи онтологий позволяет получать Wiki-системы с хорошей структурой и согласованной системой понятий.

Литература

1. Leuf B., Cunningham W. The Wiki Way: Quick Collaboration on the Web. Addison-Wesley Professional, 2001. 435 p.
2. Baader F., Nutt W. Basic Description Logics. In the Description Logic Handbook, edited by Baader F., Calvanese D., McGuinness D.L., Nardi D., Patel-Schneider P.F., Cambridge University Press, 2002.
3. Муромцев Д.И., Горовой В.А., Малинин А.А., Гаврилова Т.А., Злобин А.Н., Катков Ю.В. Интеграция wiki-технологии и онтологического моделирования в задаче управления знаниями предприятия // Труды 11-ой национальной конференции по искусственному интеллекту с международным участием КИИ-2008 (г. Дубна, Россия). — М.: ЛЕНАНД, 2008. — Т.3. — С. 360-368.
4. Режим доступа: <http://mediawiki.org> — MediaWiki.
5. Режим доступа: <http://semantic-mediawiki.org> — Semantic MediaWiki.
6. Режим доступа: http://semantic-mediawiki.org/wiki/Help:Ontology_import — описание импорта в Semantic MediaWiki.
7. Backhaus M., Kelso J., Bacher J., Herre H., Hoehndorf R., Loebe F., Visagie J. BOWiki – a collaborative annotation and ontology curation framework // In Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge, CKC Banff, Canada, May 8, of CEUR Workshop Proceedings. Volume 273. Edited by Noy N., Alani H., Stumme G., Mika P., Sure Y.,

Vrandecic D. Aachen, Germany: CEUR-WS.org, 2007.

8. Herre H., Heller B., Burek P., Hoehndorf R., Loebe F., Michalek H. General Formal Ontology (GFO) – A foundational ontology integrating objects and processes [Version 1.0] // Onto-Med Report 8, Research Group Ontologies in Medicine, Institute of Medical Informatics, Statistics and Epidemiology, University of Leipzig, Leipzig (2006).

9. Режим доступа: <http://www.mediawiki.org/wiki/Extension:RDFIO> — RDFIO.

10. Режим доступа: <http://www.w3.org/RDF/> — RDF.

11. Режим доступа: <http://www.mediawiki.org/wiki/Extension:LinkedWiki> — LinkedWiki.

12. Режим доступа: <http://pywikipediabot.sourceforge.net> —

Python WikipediaBot Framework.

13. Режим доступа: <http://protege.stanford.edu> — Protégé.

14. Motik B., Patel-Schneider P.F., Parsia B., eds. OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax // W3C Recommendation, 27 October 2009, <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/>.

15. Режим доступа: <http://rdflib.net> — RDFLib.

16. Völkel M., Krötzsch M., Vrandecic D., Haller H. and Studer R. Semantic Wikipedia // In Proceedings of the 15th International Conference on World Wide Web (Edinburgh, Scotland, May 23–26, 2006). WWW '06. ACM Press, New York, NY, pp. 585–594, 2006, <<http://doi.acm.org/10.1145/1135777.1135863>>.

Шестаков В.К. Автоматическая генерация структуры Wiki-систем при помощи онтологий. В статье описываются инструментальные средства, поддерживающие технологию автоматической генерации структуры систем на основе Wiki-технологии при помощи онтологий предметных областей. Технология позволяет получать Wiki-системы с хорошей структурой и согласованной системой понятий.

Ключевые слова: онтология, Wiki, семантические технологии, генерация структуры информационных систем.

Shestakov V.K. Automatic ontology-based generation of Wiki-system structure. The paper presents tools supporting technology of an automatic generation of structure of systems based on Wiki-technology using subject domain ontology. The technology allows to create Wiki-systems with a good structure and a consistent system of concepts.

Keywords: ontology, Wiki, semantic technology, generation structure of information system.

Статья поступила в редакцию 20.11.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

CONTENT

Andryukhin A.I., Reflective Boolean functions.....	4
Belkov D.V., Edemskay E.N. Analysis of UDP-traffic.....	14
Borota A.V., Borota A.A., Kukhto A.P., Baziyan-Kukhto N.K., Fedyaev O.I., Bakalenko V.S. Development of information medical records system of the abdominal cavity of patients... ..	21
Vinogradov G.P., Vinogradova N.G. Decision making in systems with the subjective form of understanding of interests and ideas about a situation of choice.....	28
Zemlyansky S.Y., Slozhenitsyn A.V., Andrievskaya N.K. Model structure for the people group behavior forecasting	37
Kolomoitseva I.A., Zavadskaja T.V. Information Search Features in metasearch GRID-system "Programming for Android"	44
Kostyuk A.I., Lebedeva E.O. Information system for accounting staff of the organization.....	52
Medgaus Sergey, Chernyshova Alla Program realization obfuscating algorithms of JavaScript source code.....	59
Minenko A.S. Information technology in the modeling and recognition of human face by his facial photos.....	67
Morozova O.V. Response Mechanism of intellectual shell of knowledge modules	73
Pirskaya L.V., Kravchenko P.P. Zvyagintseva A.V. Research on use delta-transformations o determine the position of unmanned aircraft.	81
Fedyaev O. Formal verification cycle programs.....	89
Filer Z.E., Andryukhin AI Computer investigation and visualization of vibrational dynamics of systems in complex form.....	98
Cherkasov. V.A., Azarov V.N. Electronic education e-learning.....	109
Shestakov V.K. Automatic ontology-based generation of Wiki-system structure.	116

Научное издание

ИНФОРМАТИКА И КИБЕРНЕТИКА

Научный журнал

(на русском, украинском, английском языках)

№ 4(6)-2016

Ответственный за выпуск А.И. Андрюхин

Технические редакторы Т.В. Завадская, А.В. Звягинцева

Компьютерная верстка Д.А. Филипишин

Подписано к печати 26.12.2016. Формат 60×84 1/8. Бумага офсетная. Печать ризографическая.
Усл. печ. лист. 10. Уч.- изд. лист. 7,1. Тираж 100 экз.

Адрес редакции: ДНР, 83001, г. Донецк, ул. Артема, 58, ГБУЗ «ДонНТУ», 5-й учебный корпус,
к. 425. Тел.: +38 (062) 301-08-51 E-mail: infcyb.donntu@yandex.ru, URL: <http://infcyb.donntu.org>

Издатель Государственное высшее учебное заведение «Донецкий национальный технический
университет» ДНР, 83001, г. Донецк, ул. Артема, 58. Тел.: +38 (062) 301-08-67, +38 (062) 301-
09-67 Свидетельство о государственной регистрации субъекта издательского дела: серия ДК
№2982 от 21.09.2007

Отпечатано ООО фирма «ДРУК-ИНФО» ЛУН, 83001, г. Донецк, ул. Артема, 58, к. 113, тел.: +38
(062) 335-64-55