

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ**



ИНФОРМАТИКА И КИБЕРНЕТИКА

3(9)

Донецк – 2017

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

ИНФОРМАТИКА И КИБЕРНЕТИКА, № 3(9), 2017,
Донецк, ДонНТУ.

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 ГОУ ВПО «Донецкий национальный технический университет» Министерства образования и науки ДНР. Протокол №. 9 от 27.10.2017.

Журнал зарегистрирован в Министерстве информации ДНР.

Свидетельство о регистрации: серия ААА № 000145 от 20.06.2017.

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, 2017

СОДЕРЖАНИЕ

| | |
|--|-----|
| Вероятностные автоматы и асинхронность Андрюхин А.И | 5 |
| Обобщения закона Мура Аноприенко А.Я..... | 14 |
| Подсистема оптимизации работы гибких производственных систем с использованием многокритериальных генетических алгоритмов Бабич К. К., Секирин А.И., Новиков Д.Д..... | 24 |
| Вычислительные структуры для размещения файлов по узлам компьютерной сети Бельков Д.В | 29 |
| Модель нечеткой логики в функциональном алгоритме управления информацией и коммуникациями предприятий Боднар А.В., Николаенко Д.В..... | 37 |
| Применение критериев принятия решений и теории игр при управлении экономическими показателями металлургического предприятия В.А. Гуляев, Ю.К. Орлов..... | 43 |
| Архитектура и сценарии распределённой обработки медицинских изображений веб-ориентированной PACS Двойкин А. А., Привалов М. В., Воронова А. И. | 50 |
| Применение методов помехоустойчивого кодирования для компактного тестирования цифровых схем О.Н. Дяченко, Ю.Е. Зинченко, В.О. Дяченко..... | 55 |
| Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам. Зинченко Ю. Е., В. С. Ленч, Т. А. Зинченко, В.Н. Павлыш | 60 |
| Проектирование компьютерных компонентов для реализации операций тетралогии С. В. Иваница..... | 69 |
| Модель поддержки принятия решений в управлении жизненным циклом мобильного приложения Искра Е.А..... | 77 |
| Исследование графической системы с вертикальной архитектурой для поиска пересечения луча с группой полигональных объектов Р.В. Мальчева, Т.В. Завадская, В.И. Кудояр..... | 83 |
| Анализ методов распознавания эмоционального состояния человека для выявления аномального поведения в видеоаналитических системах Т. В. Мартыненко, А.А. Щербаков, Б.В. Боднар..... | 89 |
| Исследование проблемы выбора минимального набора услуг для расширения деятельности предприятия на примере интернет-провайдера И.В. Матях, Е.О. Савкова, О.В. Чен..... | 94 |
| Приближенный анализ конвективной задачи Стефана. Миненко А.С., Радевич Е.В..... | 100 |
| Обучающая система работы с контейнерными классами. Ольшевский А.И. | 106 |

| | |
|---|-----|
| Разработка функциональной структуры логистической системы формирования заказов для интернет-магазина В.А. Светличная, Н.К. Андриевская, К.Ю. Чаленко | 111 |
| Разработка системы контроля и мониторинга состояния охраны труда и промышленной безопасности на угледобывающих предприятиях. Тарасова И.А., Резникова Г.А..... | 119 |
| Content..... | 128 |

УДК 004.0-519.854.001

Вероятностные автоматы и асинхронность

А.И. Андрухин, к.т.н,
Донецкий национальный технический университет, г. Донецк,
alexandruckin@ramber.ru

Андрухин А.И. Вероятностные автоматы и асинхронность. Целью исследования является применение моделей вероятностных автоматов для повышения точности описания асинхронных процессов. Оно основано на анализе неопределенности начального состояния и предположениях относительно длительности переходов в вероятностных автоматах. В работе была выполнена разработка алгоритма статистического моделирования для оценок характеристик асинхронных переходов в вероятностных автоматах. Приведены результаты компьютерных расчетов. В расчетах использовался пакет Mathematica.

Ключевые слова: вероятностный автомат, асинхронность, компьютерная модель, функция распределения.

Введение

В стандартной классификации математических моделей систем, если при исследовании случайные факторы не учитываются, мы имеем дело с детерминированными моделями. Для описания систем, функционирующих в непрерывном времени, применяем дифференциальные, интегральные и другие уравнения, а для описания систем, действующих в дискретном времени, - конечные автоматы и конечно-разностные схемы. Если же при исследовании мы случайные факторы учитываем, то используем стохастические модели, т.е. для представления систем с непрерывным временем применяем модели систем массового обслуживания (СМО), а для описания систем с дискретным временем используем вероятностные автоматы (ВА).

Согласно [1-8], основой описания вероятностных автоматов является вероятностная функция, описывающая его поведение $\text{Prob}[s(t+1)|y(t), x(t)]$, которую обычно разбивают на две части:

1) Функцию перехода автомата из одного состояния в другое.

2) Выходную функцию.

Для целей статьи достаточно использовать табличное представление переходов вероятностных автоматов и таблицу выходов.

Эти таблицы переходов и выходов ВА задают вероятности перехода ВА в состояние $s(t+1)$ в момент времени $t+1$ и выдачи выходного сигнала $y(t)$ в момент времени t в зависимости от нахождения в состоянии $s(t)$ и воздействия $x(t)$ в момент времени t .

Целью исследования является применение моделей ВА для описания при повышении точности асинхронных процессов. Оно основано на анализе неопределенности

начального состояния и предположениях относительно длительности переходов в ВА.

Задачей исследования является разработка алгоритма статистического моделирования для оценок характеристик асинхронных переходов в ВА.

Более конкретно, выполняются расчеты для определения доли вероятностных автоматов с задаваемыми распределениями входов, выходов состояний, которые :

1) удовлетворяют условию псевдопоглощения, т.е. они постоянно выдают конкретный выход Y_x при переходах по различным состояниям. Напомним, что условие поглощения определяется выражением $\delta(S,X)=\delta(\delta(S,X),X)$, т.е. постоянное воздействие одних и тех же входных сигналов в автоматах должно приводить их в стационарное состояние.

Одной из важнейшей областью применения ВА является их применение при описании работы процессора(ов) ЭВМ в многозадачном режиме, т.е. при описании асинхронных параллельных процессов.

Для детерминированных автоматов при их реализации чрезвычайно важен учет асинхронности переходов, которые обусловлены состязаниями сигналов [9].

Состязания элементов памяти приводят к тому, что автомат при изменении состояния не сразу оказывается в том состоянии, которое запланировано требованиями проекта, а переходит в него через несколько непредусмотренных транзитных состояний. Если в результате такого перехода, независимо от соотношений задержек элементов памяти, автомат достигает того состояния, в которое он должен перейти, то такие состязания считаются некритическими. Если же существует хотя бы одна комбинация значений задержек элементов памяти, при которой автомат не достигает фиксированного требуемого

состояния, то такие состязания являются существенными или критическими.

Согласно [9], присутствие существенных состязаний можно определить по таблице переходов на уровне абстрактного автомата. Рассмотрим соседнее изменение входов $X_K \rightarrow X_L$ на рис.1.

| | | |
|-------|-------|-------|
| | X_K | X_L |
| S_I | S_I | S_J |
| S_J | S_M | S_J |
| S_M | S_M | S_P |

Рисунок –1.Фрагмент таблицы переходов.

Автомат должен перейти в устойчивое состояние S_J .

Однако структурная реализация может привести к тому, что новое состояние S_J воспринимается частью схемы при старых входных воздействиях X_K . Поэтому возможен переход схемы в состояние S_J . Полностью завершившийся переход $X_K \rightarrow X_L$ может привести к переходу всей схемы из состояния S_M в состояние S_P .

Некритические состязания могут существенно изменять время, затрачиваемое автоматом на переход в нужное состояние, поскольку автомат может совершать различное число транзитных переходов в зависимости от соотношения между задержками.

Основные обозначения для модели расчета

Обозначим через nX , nY , nS число входов, выходов и состояний исследуемых ВА. Введем следующие обозначения.

Пусть $S_{ikj} = \text{Prob}(s=i, x=k, s_{\text{new}}=j)$ - вероятность перехода ВА из состояния $s=i$ в состояние $s_{\text{new}}=j$ при входном воздействии $x=k$ и $Y_{ikl} = \text{Prob}(s=i, x=k, y=l)$ - вероятность выдачи ВА выходного сигнала $y=l$ из состояния $s=i$ при входном воздействии $x=k$.

Для ВА выполняются следующие соотношения

$$\sum_{i=1}^{nS} \sum_{j=1}^{nS} S_{ikj} = 1$$

$$\sum_{i=1}^{nS} \sum_{l=1}^{nY} Y_{ikl} = 1$$

Напомним, что при проектировании обычно учитывают условие поглощения $\delta(S,X) = \delta(\delta(S,X),X)$, т.е. постоянное воздействие одних и тех же входных сигналов для технических устройств должно приводить их в стационарное состояние.

Математическая модель ВА для расчета

Основываясь на рассуждениях для асинхронных конечных детерминированных автоматов, построим оценки по аналогии для ВА.

Рассмотрим рис. 2, 3.

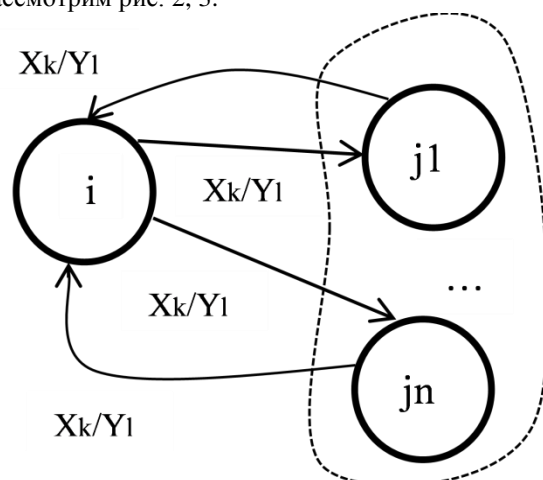


Рисунок 2-Граф переходов при P=2 .

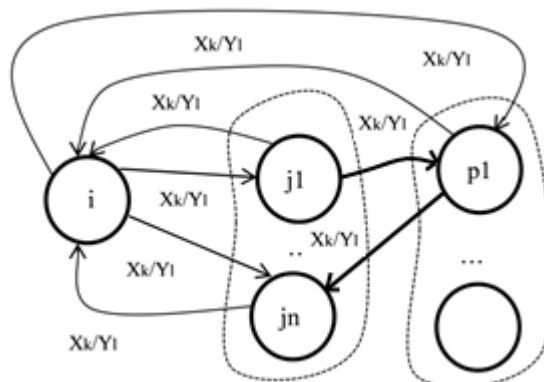


Рисунок 3-Граф переходов при P=3.

Выполним оценку вероятности Y_{2ikl} нахождения в состоянии i при дальнейшем переходе в состояния $j1, \dots, jn$ с возвратом в i (т.е. за 2 перехода). Для этого мы используем формулу

$$Y_{2ikl} = \sum_{j=1}^{nS} S_{ikj} Y_{ikl} S_{jki} Y_{jkl}$$

Аналогично построим оценку вероятности Y_{3ikl} нахождения в состоянии i при дальнейшем переходе в состояния j_1, \dots, j_n и далее при переходе из этих состояний в состояния p_1, \dots, p_n с возвратом в i (т.е. за 3 перехода).

Аналогичная формула представлена ниже

$$Y_{3ikl} = \sum_{p=1}^{nS} \sum_{j=1}^{nS} S_{ikj} Y_{ikl} S_{jkp} Y_{jkl} S_{pki} Y_{pkl}$$

Алгоритм расчета

Считая, что условия неопределенности имеют случайную природу, мы фиксируем данные по наблюдаемым выходным характеристикам, т.е. таблицу Y распределения выходов ВА.

Тогда алгоритм представим в следующем виде:

1. Установка номера распределения $I=0$.
2. $I=I+1$.
3. Если $I>I_{max}$, то конец расчета.
4. Начало цикла по испытаниям: $Z=0$, $S=0$.
5. $Z=Z+1$.
6. Если $Z>Z_{max}$, то переход на п.9.
7. Заполнение массива S .
8. Переход на п.5
9. Вычисление $S_k = \sum_i \sum_j S_{ikj}$.
10. Нормирование $S_{ikj} = S_{ikj} / S_k$.
11. Обработка и выдача результатов по распределению I .
12. Переход на п.2 (новое распределение)

На рис.4 представлены результаты расчетов для равномерного распределения.

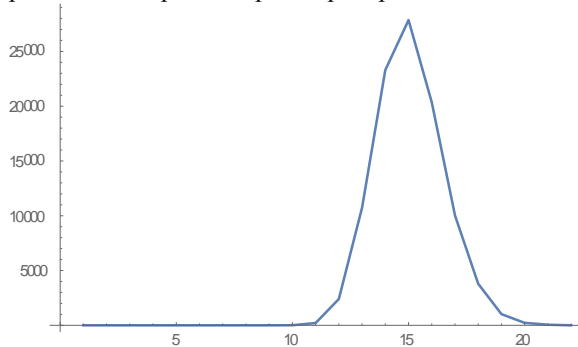


Рисунок 4-Гистограмма частоты $Z_{max}=100000$.

Основная схема экспериментальных расчетов

На рис.5 представлена матрица марковской цепи для анализа соответствующего ВА. Количество строк этой матрицы равно количеству состояний NS умноженное на количество входных сигналов NX .

Количество столбцов этой матрицы равно произведению числа состояний NS и количества выходных сигналов NY . Элементом матрицы является значение вероятности P_{ij} перехода из состояния i в состояние j , где i определяется парой (s,x) , а j соответствует паре (s,y) . Здесь $s=1, NS$ – состояние ВА, $x=1, NX$ – входной сигнал, а $y=1, NY$ – выходной сигнал ВА.

| | | | | | | | | |
|-----|-----|-----|-----|----------|----|-----|-----|-----|
| | 1,1 | 1,2 | ... | K,1 | .. | S,1 | ... | S,Y |
| 1,1 | | | | | | | | |
| 1,2 | | | | | | | | |
| ... | | | | | | | | |
| I,1 | | | | | | | | |
| I,2 | | | | P_{ij} | | | | |
| ... | | | | | | | | |
| S,1 | | | | | | | | |
| S,2 | | | | | | | | |
| ... | | | | | | | | |
| S,X | | | | | | | | |

Рисунок 5-Модель марковской цепи для анализа вероятностного автомата.

Алгоритм определения вероятностей определенной длины (цепочки) переходов при постоянном воздействии фиксированного сигнала x можно представить согласно рис.6 следующим образом.

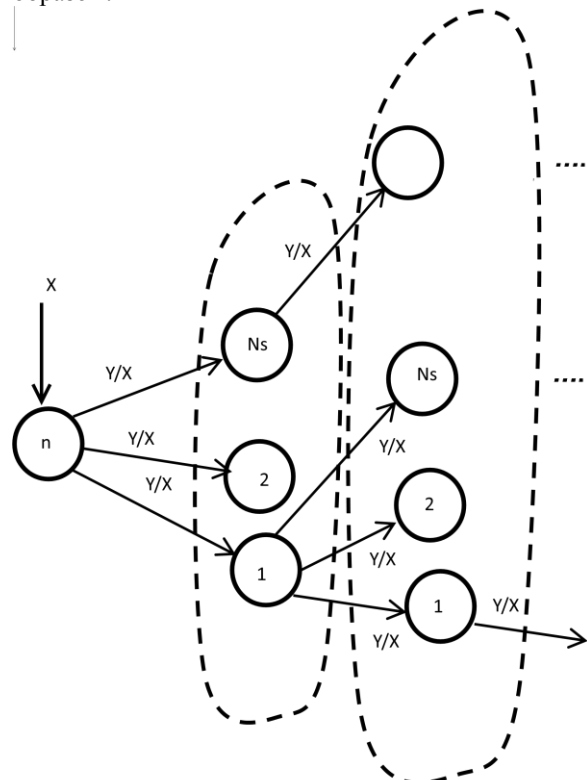


Рисунок 6-Расчет вероятностей для переходов различной длины в ВА.

Для фиксированной пары (s, x) , в которую мы попали за $i-1$ переход, существует NS пар – последователей (s_k, y_0) с требуемым выходом y_0 .

Можем обозначить через V_i множество переходов длины i из конкретной фиксированной пары (s, x) , которые всегда выдают определенный выходной сигнал y_0 . Таким образом, $L(i)$ - число переходов длины i равно $NS \cdot L(i-1)$ и следовательно, $L(i) = NS^i$. Элементу V_i соответствует вероятность перехода, которая равна произведению вероятностей соответствующих i переходов согласно матрице.

Представляет интерес решение вопроса, как влияет на распределение вероятностей для множеств V_i вариация числа состояний, входных и выходных реакций вероятностного автомата при определенном распределении вероятностей P_{ij} в соответствующей ВА матрице марковской цепи.

На рис.7-15 для равномерного распределения имеем результаты расчетов для определенных размерностей ВА, где усредненная реализация получена для 10000 испытаний.

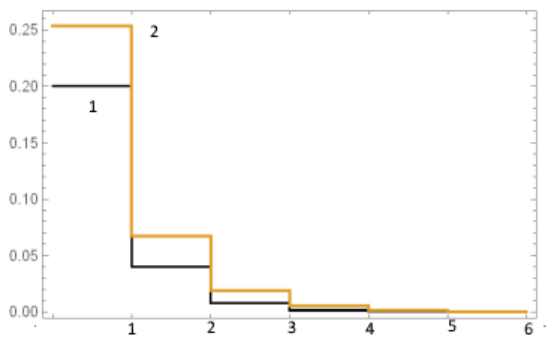


Рисунок 7-Вероятности длин переходов при $s=6, y=4, x=2$:1-реализация, 2-усредненное среднее.

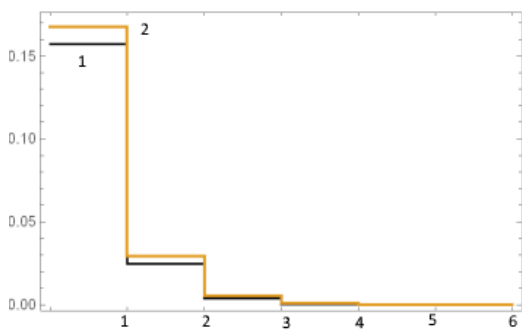


Рисунок 8-Вероятности длин переходов при $s=6, y=6, x=2$:1-реализация, 2-усредненное среднее.

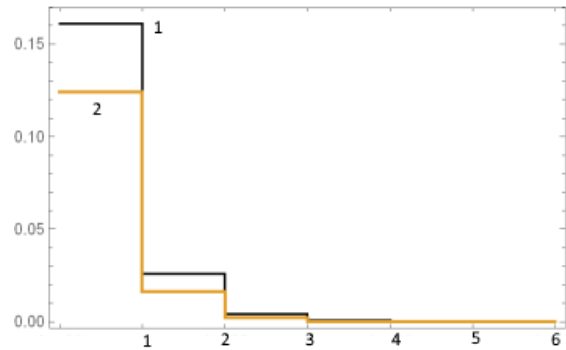


Рисунок 9-Вероятности длин переходов при $s=6, y=8, x=2$:1-реализация, 2-усредненное среднее.

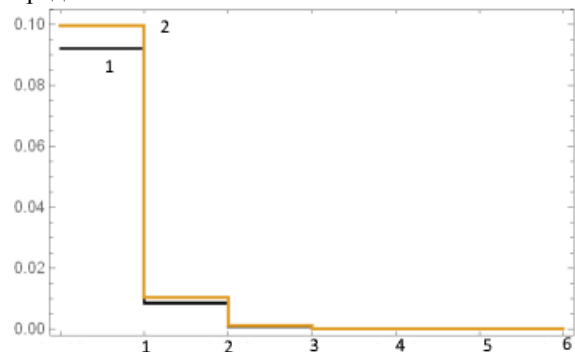


Рисунок 10-Вероятности длин переходов при $s=6, y=10, x=2$:1-реализация, 2-усредненное среднее.

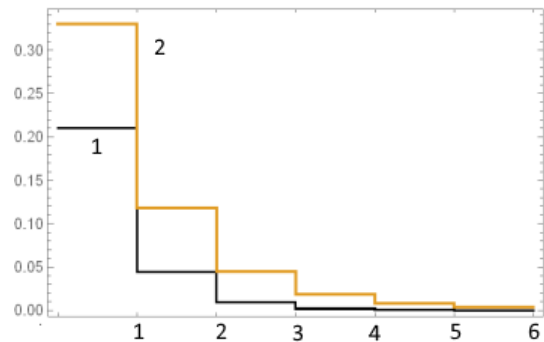


Рисунок 11-Вероятности длин переходов при $s=4, y=3, x=2$:1-реализация, 2-усредненное среднее.

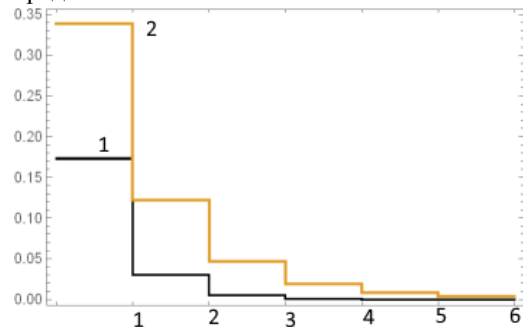


Рисунок 12-Вероятности длин переходов при $s=5, y=3, x=2$:1-реализация, 2-усредненное среднее.

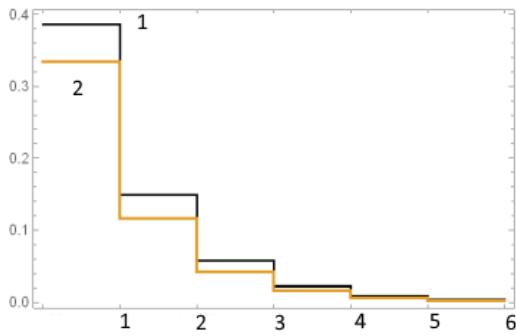


Рисунок 14-Вероятности длин переходов при $s=7, y=3, x=2$:1-реализация, 2-усредненное среднее

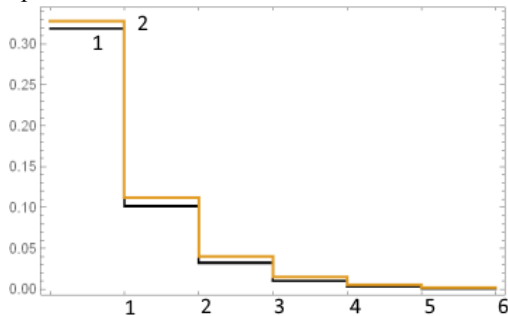


Рисунок 15-Вероятности длин переходов при $s=8, y=3, x=2$:1-реализация, 2-усредненное среднее.

Оценки времени перехода для ВА

Общая модель ВА не включает в себя временные параметры, но естественно большое количество специалистов конкретизировали распределения времен перехода в различных проблемных областях и предлагали решения [10-13]. Так укажем важную модель синхронизации (корреляции) функционирования различных иерархических уровней головного мозга, которая исследовалась в [13]. В ней использовалось геометрическое распределение.

Интерес представляют такие вероятностные распределения времен перехода как нормальное, экспоненциальное и распределения Релея, Накагами и Гумбеля. Последнее применяется в теории надежности для описания наибольших или наименьших значений в последовательности случайных величин.

Распределения Релея и Накагами часто используется как модель для статистических сигналов, переданных через радиоканалы. Более конкретно, распределения Релея (Накагами) применяются для описания статистики флуктуаций сигнала на выходе многопутевого канала (каналов) с замираниями соответственно.

На рис.16-23 представлены результаты расчетов для различных распределений и конкретного числа переходов и x_i -длительности переходов.

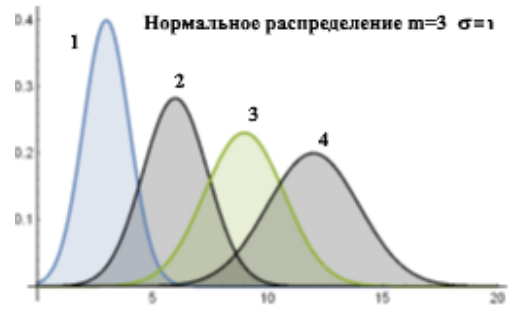


Рисунок 16-Функции плотности распределения:1- x_1 , 2- x_1+x_2 , 3- $x_1+x_2+x_3$, 4- $x_1+x_2+x_3+x_4$.

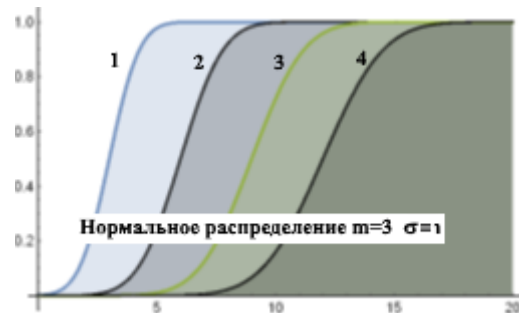


Рисунок 17-Функции распределения:1- x_1 , 2- x_1+x_2 , 3- $x_1+x_2+x_3$, 4- $x_1+x_2+x_3+x_4$.

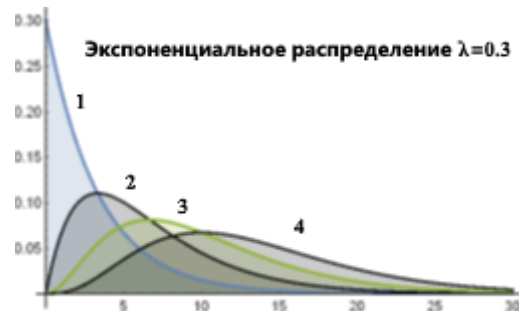


Рисунок 18-Функции плотности распределения:1- x_1 , 2- x_1+x_2 , 3- $x_1+x_2+x_3$, 4- $x_1+x_2+x_3+x_4$.

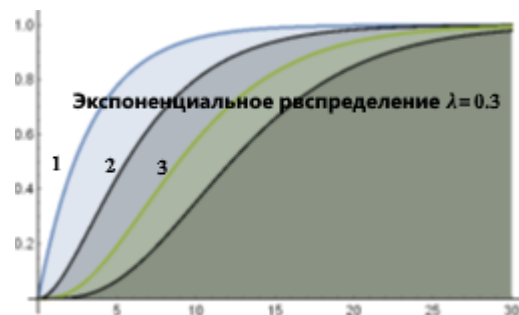


Рисунок 19-Функции распределения:1- x_1 , 2- x_1+x_2 , 3- $x_1+x_2+x_3$, 4- $x_1+x_2+x_3+x_4$.

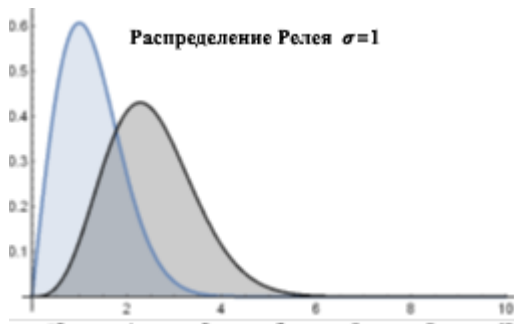


Рисунок 20-Функции плотности распределения: 1- x_1 , 2- x_1+x_2 .

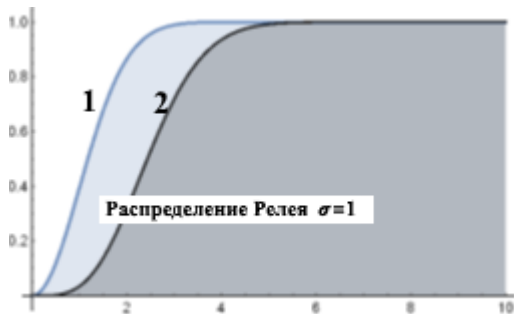


Рисунок 21-Функции распределения: 1- x_1 , 2- x_1+x_2 .

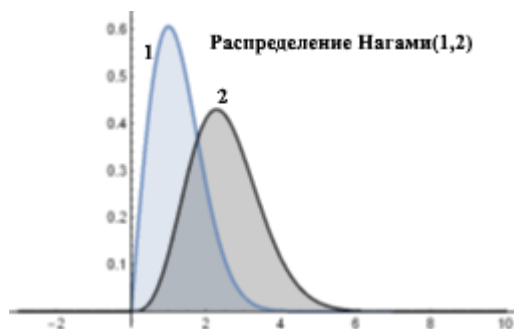


Рисунок 22-Функции плотности распределения: 1- x_1 , 2- x_1+x_2 .

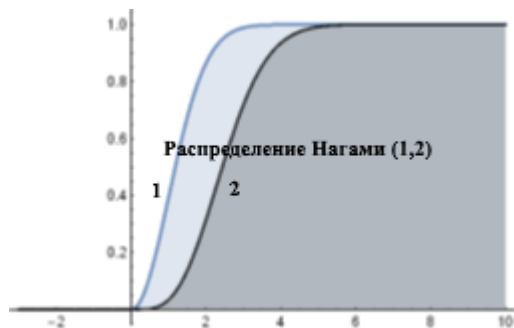


Рисунок 23-Функции распределения: 1- x_1 , 2- x_1+x_2 .

Асинхронная логика

Построение и проектирование дискретных систем и в частности цифровых устройств, которые работали бы независимо от значения времени переходов, привело к созданию различных вариантов асинхронной логики [10-12].

Общеизвестны преимуществ асинхронных схем - быстрдействие, независимость от рассинхронизации часов, низкое энергетическое рассеяние, модульная конструкция, иммунитет к метастабильному поведению, а также низкой чувствительности к электромагнитным помехам и малое их порождение [12].

Однако сложность проектирования асинхронных схем является одной из основных причин доминирования синхронных схем.

Формализмы, использованные в асинхронном проектировании интегральных схем, могут быть разделены на два класса: формализм, базирующийся на булевой алгебре и формализм, основанный на последовательностях событий. На практике большинство методологий разработки асинхронных схемах используют смешение обоих формализмов.

Разработка многих асинхронных схем основана на булевой алгебре или ее производной теории переключений. Такие схемы используют модель ограниченных задержек, и примитивные элементы - вентили, соответствующие основным логическим функциям, как И, ИЛИ, НЕ. Этот формализм удобен для реализации логических функций, анализа схем на присутствие паразитных импульсов, синтеза схем [14].

Событийный формализм имеет дело с последовательностями событий, а не переменных бинарной логики. Схемы, разработанные с помощью этого формализма и использованием модели неограниченных задержек, обычно работают в режиме ввода – вывода и имеют такие примитивные элементы, как разветвления, переключатели, и слияния. Формализм на базе событий очень удобен для конструирования асинхронных схем, при высокой степени запутанности параллелизма.

Были созданы несколько инструментальных средств для автоматической проверки асинхронных схем с событийным формализмом. Примерами таких формализмов являются, к примеру, сети Петри.

На рис.25 показано несколько простых примитивов и их обозначений, используемых в event-based методах проектирования.

Самый простой примитив *WIRE* (ПОТОК ДАННЫХ) - это двухполюсный элемент, который формирует выходное событие на выходе b после каждого входного события на входе a . События входа и выхода в потоке данных, должны чередоваться. Входное событие должно

сопровождаться выходным событием b прежде, чем произойдет другое событие. Поток данных физически осуществляется с помощью провода (шины), а события - изменениями напряжения. Инициализированный поток данных, или *IWIRE*, очень похож на *WIRE*, за исключением того, что *IWIRE* формирует событие выхода b вместо того, чтобы принять входное событие a ; после этого, его режим работы совпадает с *WIRE*.

Примитив для синхронизации – *JOIN* (объединение), также называемый *RENDEZVOUS* (рандеву). *JOIN* имеет два входа a и b и один выход c . *JOIN* выполняет операцию Логическое И двух событий – a и b . Он формирует событие выхода c только после того, как получают события оба входа, a и b . Входные данные могут измениться снова, только после того, как сформированы выходные данные. *JOIN* может быть реализован *C*-элементом Мюллера, который рассматривается далее.

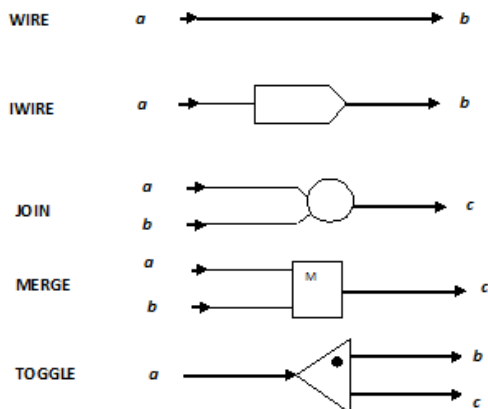


Рисунок 25-Основные нечувствительные к задержкам примитивы

Элемент *MERGE* (объединение) выполняет операцию Логическое ИЛИ двух событий. Если элемент *MERGE* получает событие на любом из его входов, a или b , он формирует событие выхода c . После входного события следует выходное; последовательные входные события не допускаются. *MERGE* может быть реализован логическим элементом *XOR*.

TOGGLE (переключатель) имеет один вход и два выхода b и c . После появления события на входе a , оно формируется на выходе b . Следующее событие на входе a формируется на выходе c . Входное событие должно сопровождаться событием выхода прежде, чем произойдет другое входное событие. Таким образом, события выхода чередуются или переключаются после каждого входного события. Точка на рисунке схемы показывает какой выход формирует событие первым.

C-элемент Мюллера назван по имени его изобретателя Д. Е. Мюллера [15]. Традиционно, его логическое поведение описывается следующим образом. Если оба входа – 0 (1), то выход становится 0 (1); иначе говоря, сохраняет то же значение. Для правильности операций *C*-элемента, также предполагается, что, как только оба входа становятся 0 (1), они не изменяются снова, пока не поменяется выход. Поведение выхода c *C*-элемента выражается с использованием значений входов a и b и предыдущего состояния выхода c' следующей логической функцией $c = c'(a \vee b) \vee a \vee b$.

C-элемент может использоваться для реализации *JOIN*, который имеет несколько более ограниченное поведение окружения, в том смысле, что входные данные не разрешено изменять дважды в последовательности. Мы приводим две самые распространенные КМОП реализации из их множества на рис. 26.

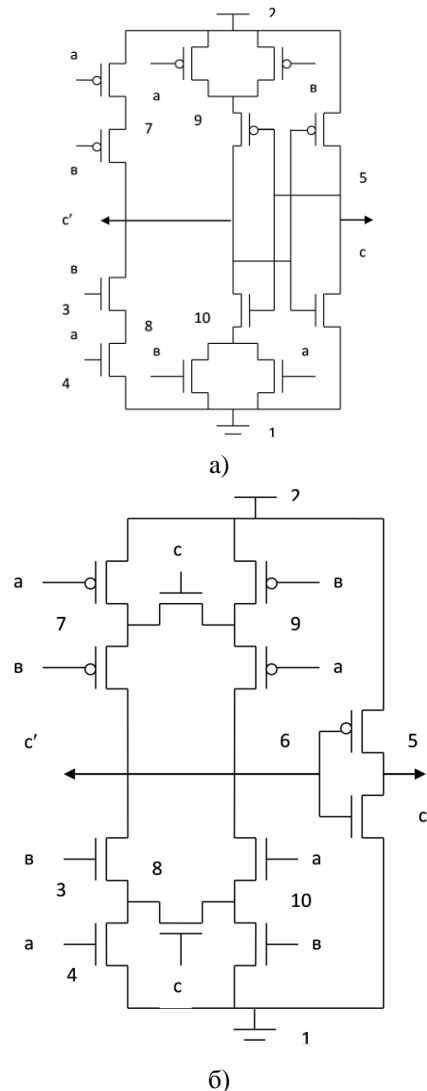


Рисунок 26-КМОП-реализации *C*-элемента:(а)-стандартная;(б)-симметрическая.

Реализация (a) – стандартная, предложенная Сазерлендом [12]. Реализация (b) предложена Ван Беркелем [16].

Каждая реализация имеет свои собственные характеристики. Реализация (b) – считается реализацией лучшей по критерию быстродействия и эффективности.

Рассмотрим два контролируемых событиями фиксатора Сазерленда [62], которые изображены на рис.27. Их работа управляется с помощью двух входных управляющих сигналов: фиксирование и проход, помеченных соответственно *c* и *p*. Они, также, имеют два управляющих выходных сигнала: осуществлено фиксирование- *cd*, и осуществлен проход- *pd*. Входные данные - *D*, выходные данные - *Q*. Реализация (a) состоит из трех так называемых переключателей на два направления. Реализация (b) включает *MERGE*, *TOGGLE* и контролируемый уровень фиксатор, состоящий из переключателя на два направления и инвертора. Переключатель на два направления схематично представлен инвертором и переключателем. Переключатель переключает между двумя позициями, в зависимости от логического значения управляющего сигнала.

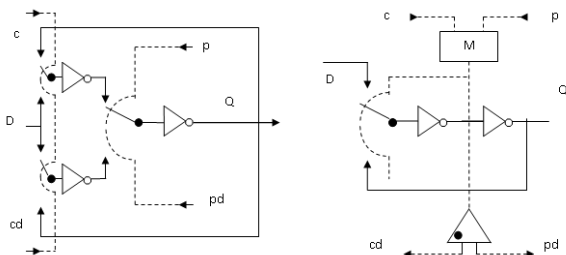


Рисунок 27-Две реализации управляемого событиями фиксатора

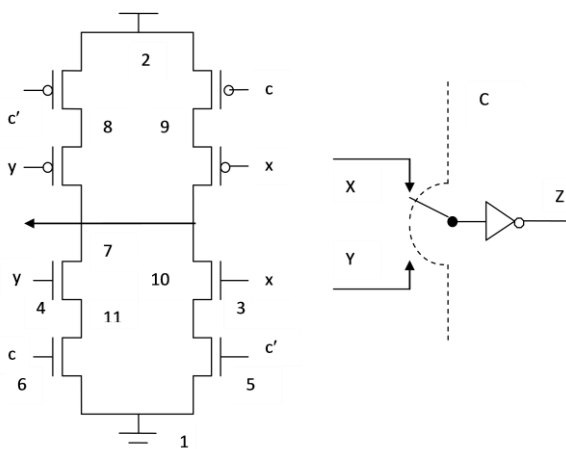


Рисунок 28-. КМОП реализация переключателя на два направления

Переключатель на два направления, фактически, является мультиплексором с двумя входами, который формирует инвертированный вариант его входного сигнала. КМОП реализация переключателя на два направления показана на рис.28 [62]. Позиция коммутатора соответствует состоянию, при котором значение входа *c* низкое.

Контролируемый событиями фиксатор имеет два состояния: прозрачное и непрозрачное.

В прозрачном состоянии никакие данные не фиксируются, но выходные данные копирует входные, потому что путь двух стадий инвертирования лежит между входом и выходом. В непрозрачном состоянии, этот путь построен так, что входные данные можно изменить без воздействия на выходные данные.

Ясно, что при этом текущие данные запираются при выводе. Реализации на рис.27(a), 27(b) показаны в начальных прозрачных состояниях. Фиксирование и проход сигналов в контролируемом событиями фиксаторе всегда чередуются. При переходе на *c*, фиксатор фиксирует текущие входные данные и становится непрозрачным. Следующее перемещение на *cd* - подтверждение провайдеру данных, о том, что текущие данные были зафиксированы и что входные данные могут быть безопасно изменены. Последующее перемещение на *p* возвращает фиксатор назад к его прозрачному состоянию, чтобы следующие данные прошли к выходу. Сигнал *p* подтверждается перемещением на *pd*. Обращаем внимание, что в реализации (a) рис.27, сигналы *cd* и *pd* просто запаздывают и возможно являются усиленными варианты *c* и *p*, соответственно.

Выводы

На практике обычно выполняют реализации конечных детерминированных автоматов в виде современных синхронных процессоров. Значительно реже аппаратные реализации автоматов являются асинхронными схемами ввиду сложности их проектирования

Однако изменения самих аппаратных реализаций (геронтологические, флуктуации окружения и т.п.) приводит к необходимости анализа стохастических приближений детерминированных конечных автоматов и общей моделью обычно выступает ВА.

Получены результаты, когда мы приближаем детерминированные автоматы с помощью ВА, т.е. тогда, когда $S_{ikj} \rightarrow 1$ только при определенном *k* и аналогично $Y_{ikl} \rightarrow 1$. Этот подход применяли в [17]. Однако объем полученных результатов не позволяет привести их в данной работе.

Это направление перспективно для дальнейших исследований

Литература

1. Paz A. Introduction to Probabilistic Automata. New York: Academic Press, 1971. 228 p.
2. Rabin M. O. Probabilistic automata // Information and Control, 1963. Vol. 6.No. 3.P. 230-245.
3. Бухараев Р. Г. Основы теории вероятностных автоматов. Москва, Наука, 1985 г.
4. Segala R., Lynch N. A. Probabilistic simulations for probabilistic processes // Nordic Journal of Computing, 1995. Vol. 2. No. 2. P. 250-273.
5. Stoelinga M. An introduction to probabilistic automata // Bulletin of the European Association for Theoretical Computer Science, 2002. Vol. 78. P. 176-198.
7. Sokolova A., deVink E. P. Probabilistic Automata: System Types, Parallel Composition and Comparison // Lecture Notes in Computer Science, 2004. Vol. 2925. P. 1-43.
8. Darwiche A. Modeling and Reasoning with Bayesian Networks. Cambridge: Cambridge University Press, 2009. 562 p.
9. Андриюхин А.И. Моделирование и диагностирование дискретных устройств на переключательном уровне. Донецк, ГБУЗ «ДонНТУ», 2012. – 258 с.:
11. The Status of Asynchronous Design in Industry. www.scism.sbu.ac.uk/ccsv/ACiD-WG/AsyncIndustryStatus.pdf
12. I. E. Sutherland, “Micropipelines”, Communications of the ACM, Vol. 32, no. 6, pp. 720-738, June 1989.
13. Дж. Николис : Динамика иерархических систем: Эволюционное представление. М.: Мир, 1989, 490 с.
10. S. Nauck, “Asynchronous design methodologies: An overview”, Proc. IEEE, Vol. 83, No. 1, Jan. 1995, pp. 69-93.
14. J. A. Brzozowski and C.-J. Seger. Asynchronous Circuits. Springer-Verlag. 1995.
15. R. E. Miller, Switching Theory Volume II: Sequential Circuits and Machines. New York NY: John Wiley & Sons, 1965.
16. K. v. Berkel. Beware the isochronic fork. Integration, the VLSI journal, vol. 13, June 1992, pp. 103-128.
17. Поспелов Д. А. Вероятностные автоматы, М., «Энергия», 1970, 88 с.

Андриюхин А.И. Вероятностные автоматы и асинхронность. Целью исследования является применение моделей вероятностных автоматов для повышения точности описания асинхронных процессов. Оно основано на анализе неопределенности начального состояния и предположениях относительно длительности переходов в вероятностных автоматах. В работе была выполнена разработка алгоритма статистического моделирования для оценок характеристик асинхронных переходов в вероятностных автоматах. Приведены результаты компьютерных расчетов. В расчетах использовался пакет Mathematica.

Ключевые слова: вероятностные автоматы, асинхронность, компьютерная модель, функция распределения.

Andryukhin A.I. Probabilistic automata and asynchrony. The aim of the study is to apply models of probability automata to improve the accuracy of describing asynchronous processes. It is based on an analysis of the uncertainty of the initial state and assumptions about the duration of transitions in probabilistic automata. In this work, a statistical modeling algorithm was developed for estimating the characteristics of asynchronous transitions in probabilistic automata. The results of computer calculations are presented. The calculations used the Mathematica package.

Keywords: probabilistic automata, asynchrony, computer model, distribution function.

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

Обобщения закона Мура

А.Я. Аноприенко

Донецкий национальный технический университет
anoprien@ya.ru

Аноприенко А.Я. Обобщения закона Мура. Систематизированы и обобщены эмпирические наблюдения, которые характеризуют динамику экспоненциального развития компьютерных технологий. Показано, что многообразие основных закономерностей, в том числе характеризующих закон Мура, может быть представлено в виде ряда обобщенных относительно простых зависимостей. Предполагается, что выявленная система закономерностей обладает достаточными прогностическими свойствами необходимой точности на краткосрочную и долгосрочную перспективу.

Введение

Так называемый закон Мура является одной из наиболее известных закономерностей в области компьютерных наук и технологий. О ней знают практически все профессионалы в данной области, но интерпретируют ее зачастую по-разному. Связано это в первую очередь с непростой историей формирования данной закономерности и ее понимания в разные периоды развития компьютерных систем и технологий. Практически все при этом знают и помнят о том, что речь идет об удвоении. Но при уточнении того, что именно удваивается и за какой период, начинаются существенные расхождения. Кроме этого, зачастую не видят принципиальной разницы между периодами удвоения в два или полтора года.

Обыденное мышление устроено так, что рост с известным периодом удвоения с трудом экстраполируется на более длительные периоды, соизмеримые со сроком действия той или иной закономерности. В этой связи уместно в очередной раз напомнить легенду об изобретателе шахмат, который запросил за свое изобретение удивительно скромную на первый взгляд награду: он просто попросил столько зерен пшеницы, сколько получится, если на первую клетку шахматной доски положить одно зерно, на вторую – в 2 раза больше, и так далее, пока не заполнятся все 64 ячейки.

В действительности оказывается, что пшеничных зерен при этом потребуется 18 446 744 073 миллиарда общей массой около 500-т миллиардов тонн! Это почти в тысячу раз больше, чем во всем мире выращивают за год в настоящее время (только к началу 1990-х годов мировой ежегодный сбор пшеницы превысил 500 миллионов тонн, а в начале 1960-х годов ежегодные урожаи составляли всего 250 миллионов тонн), и соизмеримо с суммарным

урожаем пшеницы, собранным за всю историю человечества!

Следовательно, для более адекватного понимания динамики развития компьютерных систем и технологий необходимы оценки, ориентированные на более длительные промежутки времени, измеряемые, как правило, десятилетиями, при этом существенно более точные и гибкие, чем традиционно принятые периоды удвоения показателей. Как показал анализ наблюдавшихся ранее и наблюдаемых в настоящее время закономерностей, оптимальными являются периоды, соизмеримые с продолжительностью действия типичных закономерностей и типовыми дальними горизонтами планирования человеческой деятельности. Речь в первую очередь идет о периодах в 10-20 лет. Характерно, что за указанные периоды достаточно часто наблюдается рост на 1-3 и более десятичных порядков, что позволяет в случае продолжения действия соответствующих закономерностей достаточно просто оценивать рост и в последующие десятилетия: если, например, известно, что некоторый показатель увеличивается примерно в 100 раз за 10 лет, т.е. на 2 порядка, то совсем несложно определить, что через 20 лет рост будет уже на 4 порядка, а через 30 – на 6. Если же мы знаем, что за 18 месяцев происходит удвоение, то оценка роста за 10 или, тем более, 20 лет будет существенно сложнее. Для этого, как минимум, потребуется умение быстро оценивать количество периодов удвоения и затем возводить 2 в требуемую степень. Кроме этого, важно также помнить, что незначительные на первый взгляд погрешности в оценках роста за относительно короткие периоды (за несколько лет) приводят к весьма существенным расхождениям в более длительной перспективе (десятки лет). При этом заметное влияние на эти оценки будут также оказывать неизбежные

кратковременные колебания темпов роста в пределах нескольких лет. Если же ориентироваться на 10-летние и более продолжительные периоды, то мы получаем возможность вычислить намного более точные оценки и максимально исключить влияние относительно случайных факторов, влияющих на темпы роста в те или иные годы.

Далее представлен обзор выявленных на сегодня эмпирических закономерностей роста в области компьютерных наук и технологий, прямо или косвенно связанных с законом Мура (почти все они носят экспоненциальный характер), и предложены 4 варианта обобщения данных закономерностей.

Закон Мура и его аналоги

Одним из отражений стремительного развития компьютерных технологий явилось появление множества именных (названных по именам их авторов и/или исследователей) эмпирических законов, характеризующих развитие различных технологий эпохи информационно-компьютерной революции. В первую очередь речь идет о самом законе Мура и непосредственно связанных с ним закономерностях. Краткий (и весьма неполный) их перечень включает, в частности, следующие закономерности:

Закон Мура (Moore's Law) – эмпирическое наблюдение, изначально (в 1965 г.) сделанное Гордоном Муром, согласно которому количество транзисторов, размещаемых на кристалле интегральной схемы, удваивается каждый год. В 1975 году откорректировал свое наблюдение и предсказал удвоение степени интеграции микросхем каждые 2 года. В последующем появились и другие интерпретации данного закона. В настоящее время основным вариантом интерпретации закона Мура является удвоение каждые 18 месяцев.

Закон Рока (Rock's law) или второй закон Мура, сформулированный в середине 90-х годов Юджином Мейераном и утверждающий, что стоимость фабрик по производству полупроводников аналогично закону Мура удваивается примерно каждые 4 года.

Закон Куми (Koomey's Law) – своеобразный «экологический» вариант закона Мура, гласящий, что фундаментальной особенностью развития вычислительной техники является рост энергоэффективности (т.е. среднего количества вычислений на единицу электроэнергии), возрастающий примерно в два раза каждые полтора года.

Закон Крайдера (Kryder's Law) – вариант закона Мура для дисковых накопителей,

предложенный вице-президентом по научным разработкам компании Seagate в 2005 г. и констатирующий, что плотность записи на магнитные диски удваивается приблизительно каждые восемнадцать месяцев. Это также означает, что стоимость хранения информации снижается вдвое каждые восемнадцать месяцев.

Отдельная группа закономерностей связана с ростом производительности компьютерных систем и эффективности программного обеспечения:

Закон Гроша (Grosch's Law), сформулированный еще в 1965 году, предполагающий, что производительность компьютеров, увеличивается как квадрат их стоимости. В действительности это правило действует далеко не всегда и ограничивается, как минимум, **законом Амдала** (Amdahl's law), сформулированным несколько позже, в 1967 году, и устанавливающим верхний предел роста производительности при распараллеливании вычислений. В первоначальной формулировке этот закон выглядит примерно следующим образом: «В случае, когда задача разделяется на несколько частей, суммарное время ее выполнения на параллельной системе не может быть меньше времени выполнения самого длинного фрагмента». В дальнейшем, в 1988 году, последовало уточнение в виде **закона Густафсона-Барсиса** (Gustafson – Barsis's law), определяющего оценку максимально достижимого ускорения выполнения параллельной программы в зависимости от количества одновременно выполняемых потоков вычислений («процессов») и доли последовательных расчётов в соответствии со следующей формулой для оценки ускорения масштабирования (англ. scaled speedup):

$$S_p = p + (1 - p)g, \text{ где}$$

p – количество процессоров,

g – доля последовательных расчётов в программе.

Закон Вирта (Wirth's law) – полушутливое утверждение, сделанное в 1995 году, что программы становятся медленнее более стремительно, чем компьютеры становятся быстрее.

Закон Гейтса (Gates' law) – вариант закона Вирта, названный в честь основателя Microsoft Билла Гейтса. Это также полушутливое наблюдение, утверждающее, что скорость программного обеспечения уменьшается на половину каждые полтора года, что сводит на нет все преимущества закона Мура. Основные причины: добавление избыточных ненужных функций, плохой код, нежелание программистов дорабатывать программы, плохой менеджмент

или частая смена команды.

Закон Мэя (May's law) – аналогично закону Гейтса гласит, что эффективность программного обеспечения падает вдвое каждые 18 месяцев, компенсируя закон Мура. Дэвид Мэй (David May), бывший архитектор семейства транспьютерных микропроцессоров (группы T2, T4, T8, T900) в знаменитой английской компании Inmos (язык Ossam), а затем – профессор Бристольского университета в Великобритании, причиной всего этого считает то, что нынешние программные содержат слишком много ошибок, а также слишком велики и сложны для понимания.

Можно также выделить несколько эмпирических закономерностей, связанных с развитием коммуникационных систем:

Закон Буттера (Butter's Law) – количество данных, передаваемых через волоконно-оптические линии связи, удваивается каждые 9 месяцев.

Закон Купера (Cooper's Law) – количество мобильных пользователей удваивается каждые 30 месяцев.

Закон Меткалфа (Metcalf's Law) – полезность сети пропорциональна квадрату численности пользователей этой сети.

Закон Нильсена (Nielsen's Law) – пропускная способность, доступная пользователям Интернет растет на 50 % ежегодно или удваивается каждые 21 месяц.

Анализ перечисленной выше совокупности известных эмпирических законов показывает, что практически все они в большей или меньшей степени связаны с законом Мура. Поэтому логично их систематизировать и обобщить таким образом, чтобы в каждом случае оценки скоростей роста и общего прогресса, определяемого действием перечисленных выше и прочих закономерностей, были достаточно наглядными и сравнимыми.

Минимальное обобщение

Так называемый закон Мура [1] приобрел сегодня статус фактически главной закономерности, определяющей технический прогресс не только в компьютерных технологиях, но и во многих других областях науки и техники. В то же время, как уже отмечалось, нарастает неоднозначность и неопределенность в понимании того, что же в действительности определяет данная закономерность. Современный диапазон интерпретаций закона Мура распространяется от наиболее широкого его толкования как практически любой формы экспоненциального развития систем до наиболее узкого определения, предполагающего исключительно удвоение ряда показателей

компьютерной техники каждые 1,5 года. Последний вариант интерпретации закона, являющийся на сегодня наиболее популярным, самим Муром, как это ни парадоксально, никогда не формулировался.

В целом можно утверждать, что в настоящее время в связи со значительным расширением использования термина «закон Мура» созрела необходимость всесторонне проанализировать все проявления данной закономерности и перейти к более точным и однозначным формулировкам и определениям.

Гордон Мур, один из основателей корпорации Интел [2], в 1965 г. впервые высказал предположение, что одним из наиболее экономически оправданных вариантов интенсивного развития цифровой микроэлектроники является ежегодное удвоение количества активных элементов на кристалле [3]. Позднее, в 70-х и 80-х годах, оценки темпов развития стали существенно более скромными. Период в полтора года или в 18 месяцев связан с прогнозами его коллег, пришедших к середине 80-х годов к выводу о том, что производительность процессоров должна удваиваться каждые 18 месяцев из-за сочетания роста количества транзисторов и быстродействия каждого из них – именно об этой версии закона, как правило, по умолчанию ведется речь, когда упоминается факт экспоненциального роста многих показателей в области компьютерных технологий.

Таким образом, необходимо четко различать как минимум два варианта закона Мура, которые можно обозначить в соответствии с годом их появления как «закон Мура 1965» и «закон Мура 1985». Целесообразно также ввести для дальнейшего использования сокращенные обозначения для различных версий данного закона, состоящие из аббревиатуры ML (от англоязычного исходного наименования данной эмпирической закономерности как Moore's law) и года появления соответствующей версии закона: ML1965, ML1985 и т.д.

Следует отметить, что при относительно небольшой на первый взгляд разнице в периодах удвоения (1 и 1,5 года) в десятилетней перспективе мы имеем соответственно рост в 1000 и 100 раз, т.е. на 3 и 2 десятичных порядка соответственно.

Характерно, что с середины XX века начали также отмечаться и различные прочие тенденции роста показателей в 10-кратном размере за 10 лет. Например, именно так оценивался рост объема инженерных работ [4, с. 8] и количества программируемых устройств, общее количество которых к концу 50-х годов достигло примерно 1-го миллиона и с того времени растет экспоненциально с 10-кратным увеличением за

10-летие [5, 6]. Условно этот вариант роста можно обозначить как «медленный закон Мура», который в настоящее время характеризует рост количества устройств, подключенных к Интернету. При приближении к 2005 году такие темпы роста обсуждались особенно активно в связи с тем, что именно с такой скоростью росла плотность энергии в микропроцессорах. В этой связи целесообразно обозначить данную закономерность как «закон Мура 2005» или ML2005.

Таким образом, можно выделить 3 основных варианта первого обобщения закона Мура (ML2005 – «медленный закон Мура», ML1985 – «современный закон Мура», и ML1965 – «быстрый закон Мура»), обозначив их соответственно через M1, M2 и M3, для которых ежегодные коэффициенты роста (ЕКР) будут соответственно 1,26, 1,59 и 2,00. Но при этом за 10-летний период коэффициенты роста будут иметь значения, равные 10-ти в соответствующей степени: 10 для M1, 100 для M2 и 1000 для M3.

Второе обобщение

Детальный анализ всей наблюдаемой в настоящее время совокупности закономерностей роста в области компьютерных технологий показывает, что описанное выше минимальное обобщение закона Мура охватывает далеко не все уже известные на сегодня закономерности.

В частности, Гордону Муру уже в 1975 году пришлось сделать существенное уточнение: в долговременной перспективе удвоение сложности интегральных схем возможно лишь каждые 2 года (целесообразно обозначить это как «закон Мура 1975» или ML1975) [7], что в дальнейшем, как показывает анализ, полностью подтвердилось.

В целом следует отметить, что примерно к середине каждого десятилетия (начиная с 1960-х годов) появлялась новая модификация закона Мура, которая существенно дополняла известные ранее эмпирические наблюдения. Данный факт позволяет говорить о своеобразном «законе развития закона Мура», суть которого заключается в том, что каждое десятилетие выявляется новая модификация эмпирической закономерности, которая дополняет предыдущие и описывает или новые особенности экспоненциального развития в области компьютерных технологий или разные темпы их развития на различных этапах эволюции. Закономерности, выявленные в 1965, 1975 и 1985 годах, описывающие относительно простые показатели, связанные с ростом степени интеграции и быстродействия цифровых микросхем, целесообразно назвать первым поколением законов Мура – именно эта группа

версий закона наиболее известна.

Последующие десятилетия развития компьютерных технологий позволили выявить аналогичные закономерности и по целому ряду других показателей, связанных с ростом сложности микросхем лишь косвенно. Эту группу будем называть вторым поколением законов Мура.

Начало второму поколению выявленных закономерностей было положено в середине 90-х годов, когда созрел так называемый «второй закон Мура», определяющий экспоненциальный рост стоимости производства микросхем по мере их усложнения в соответствии с «первым законом Мура». Гордон Мур в 1995 г. впервые достаточно убедительно показал, что дальнейший экспоненциальный рост полупроводниковой промышленности может существенно сдерживаться исходя из сугубо экономических ограничений, связанных с экспоненциальным удорожанием ростом стоимости соответствующих средств производства [8]. В частности он обратил внимание на то, что стоимость строительства новой более современной фабрики по производству микросхем удваивается примерно каждые 4 года. В последующем длительность периода удвоения стоимости фабрик в различных исследованиях уточнялась. При этом указывались периоды в 5 и 6 лет, что в конечном итоге привело к признанию того факта, что эта стоимость новых полупроводниковых производств растет примерно на порядок за 20 лет. Этот факт обозначим как «закон Мура 1995» (ML1995). Данную закономерность, как отмечалось выше, иногда также называют законом Рока (Rock's law) в честь Артура Рока, который в 1968 г. помог своими инвестициями основать корпорацию «Intel».

Примерно такими же темпами нарастали стоимость фотолитографического оборудования, объемы выпуска кремниевых пластин для производства микросхем и масштабы полупроводниковой промышленности в целом. Все это при сохранении нынешних темпов развития в ближайшие десятилетия ведет к достижению полупроводниковой промышленностью суммарного уровня производства, равного в стоимостном выражении суммарному уровню производства всех видов продукции всего мирового хозяйства [8]. Экстраполяция наблюдавшихся в середине 90-х годов тенденций приводила к выводу, что произойти это должно было примерно к 2050 г. Естественно, эта ситуация представлялась несколько абсурдной даже с учетом стремительно нарастающей доли «полупроводникового хозяйства» в мировой экономике. Поэтому единственно возможный вывод из всего этого был следующий: примерно в 20-е годы XXI столетия

действие закона Мура, в его нынешнем виде, станет невозможным по сугубо экономическим причинам. Именно на это обратил внимание Гордон Мур в 1995 г. [8]. В дальнейшем его наблюдения подтвердили и другие исследователи [9].

К 2005 г. выявились и новые фундаментальные технологические ограничения, которые существенно затрудняли дальнейший экспоненциальный рост характеристик микропроцессоров. В частности, по мере увеличения частоты синхронизации в 32-разрядных микропроцессорах плотность энергии в расчете на единицу площади кристалла с середины 80-х годов возрастала каждые 10 лет примерно на порядок, что приводило к стремительному увеличению нагрева кристаллов и катастрофическому усугублению проблем с теплоотводом. Экстраполяция этих тенденций на ближайшие десятилетия показывала неуклонное приближение температуры нагрева кристаллов к невероятным значениям, характерным, например, для рабочей зоны ядерных реакторов [10].

Ярко проявившаяся к 2005 г. закономерность роста плотности энергии в микропроцессорах (как отмечалось выше, десятикратно за десятилетие – «закон Мура 2005» или ML2005) привела к коренному пересмотру технической политики в дальнейшем развитии микропроцессорных технологий: рост частоты синхронизации практически прекратился. При этом дальнейший рост производительности начал обеспечиваться преимущественно за счет тотального распараллеливания вычислительных процессов, в частности, путем наращивания количества вычислительных ядер в процессорах и роста числа процессоров на кристалле.

С приближением 50-летия закона Мура выявилась еще одна любопытная закономерность, связанная в первую очередь с ростом производительности наиболее мощных компьютерных систем. Благодаря статистике роста производительности 500-т наиболее мощных суперкомпьютеров, собираемой, систематизируемой и регулярно публикуемой с 1993 года (список «Тор500»), выявилась закономерность, которая не укладывалась в известные до этого варианты закона Мура: производительность компьютерных систем в целом (а не отдельно процессоров) растет практически точно на порядок каждые 4 года.

Выяснилось также, что такие темпы экспоненциального роста (ранее не подтвердившиеся применительно к росту стоимости фабрик полупроводников) наблюдаются и в ряде других случаев. Например, этой закономерности подчиняется рост количества вычислительных ядер в

суперкомпьютерных системах, снижение стоимости хранения гигабайта информации на внешних носителях, глобальный ежегодный рост производства накопителей на жестких дисках, выраженный в виде их суммарной емкости, количество сетевых прикладных программных интерфейсов и т.д. [11]. Данную закономерность целесообразно обозначить как «закон Мура 2015» (ML2015). В отличие от двух других законов Мура второго поколения данная закономерность имеет более общий характер и в большинстве случаев может экстраполироваться на обозримое будущее без каких-либо существенных ограничений.

Таким образом, за 50 лет существования закона Мура выявилось не менее шести его модификаций, характеризующихся различными темпами экспоненциального роста. Такое разнообразие на первый взгляд вносит существенную путаницу в использование самого понятия «закон Мура» и настоятельно требует на текущем этапе его переосмысления и обобщения. В идеале необходимо выявить наиболее общую закономерность, связывающую все известные на сегодня варианты закона Мура в единую систему, описываемую достаточно простыми функциональными зависимостями.

В процессе уточнения реальных ЕКР и различных попыток их систематизации такая зависимость в итоге была выявлена. Добиться этого удалось в процессе упорядочивания всех известных к 2015 г. вариантов закона Мура в соответствии с возрастанием темпов экспоненциального роста. При этом максимально уточнялись значения как ЕКР, так и коэффициентов роста за различные многолетние периоды. В процессе исследований выяснилось, что коэффициенты роста для всех шести вариантов выстраиваются в единую шкалу, в рамках которой, начиная с «самой медленной» закономерности ML1995 с ЕКР, равным примерно 1,122, для каждого последующего «более быстрого» варианта закономерности наблюдается возрастание коэффициентов роста в 1,122. На основании данного наблюдения была получена следующая зависимость [11]:

$$P_i = P_0 2^{\frac{L(Y_i - Y_0)}{6}},$$

где L – коэффициент, равный порядковому номеру закономерности при их упорядочивании в соответствии с возрастанием темпов экспоненциального роста (табл. 1); Y_0 и Y_i – начальный и текущий год действия соответствующей закономерности; P_0 и P_i – значение наблюдаемого параметра в начальном и искомом году.

Символьное обозначение L в соответствии с англоязычными терминами, начинающимися на

эту букву, можно интерпретировать и как номер варианта закона (англ. Law), и как уровень (англ. Level) или скорость экспоненциального роста.

Таблица 1 – Коэффициенты роста для различных вариантов закона Мура

| L | ML | M | ЕКР | Рост за 10 лет | Рост за 20 лет |
|---|--------|----|-------|----------------|----------------|
| 1 | ML1995 | | 1,122 | 3,2 | 10^1 |
| 2 | ML2005 | M1 | 1.260 | 20 | 10^2 |
| 3 | ML1975 | | 1,414 | 32 | 10^3 |
| 4 | ML1985 | M2 | 1,587 | 102 | 10^4 |
| 5 | ML2015 | | 1,782 | 323 | 10^5 |
| 6 | ML1965 | M3 | 2,000 | 1024 | 10^6 |

Вышеприведенная зависимость обладает целым рядом интересных особенностей. В частности, за 6-летний период коэффициенты роста описываются степенями двойки и имеют значение 2^L . Особенно примечательным является тот факт, что за 20-летний период коэффициенты роста имеют значение 10^L . Т.к. в большинстве случаев действие различных вариантов закона Мура наблюдается на протяжении 20-ти и более лет, то знание этого факта позволяет наиболее просто классифицировать все закономерности роста в соответствии с шестью уровнями или скоростями экспоненциального роста, фактически определив десятичный порядок их роста за базовый 20-летний период (табл. 1).

К настоящему времени проанализированы десятки различных процессов экспоненциального роста в области компьютерных технологий. Самое удивительное заключается в том, что в большинстве случаев наблюдается довольно точное соответствие одному из 6-ти вариантов закономерности. Некоторые отклонения от точного соответствия вышеприведенной зависимости вполне естественны и иногда наблюдаются. Поэтому в наиболее общем случае должен учитываться поправочный или уточняющий коэффициент k , что приводит зависимость к следующему виду:

$$P_i = kP_0 2^{\frac{L(Y_i - Y_0)}{6}}$$

В большинстве случаев с точностью до нескольких знаков (как правило, 2-х или 3-х) после запятой мы имеем $k=1$. Но в общем случае k может принимать значения в диапазоне от 0,944 до 1,059. При этом, если $k < 1$, то соответствующий вариант закономерности целесообразно обозначать как «Lj-», например, «L4-», а если $k > 1$, то следует использовать обозначение «Lj+», например, «L4+». Следует также иметь в виду, что

в табл. 1 значения ЕКР приводятся с точностью не более 3-х десятичных знаков после запятой, но в процессе анализа и расчетов использовались значения с 10-ю десятичными знаками после запятой. В частности ЕКР для «самого медленного» варианта ML1995 для краткости принимается равным 1,122, но в действительности составляет 1,1224620483. Такое же значение имеет и «коэффициент ускорения» при переходе от одного варианта закономерности к следующему.

Некоторые из вариантов закона Мура, представленные в табл. 1, имеют свои специфические особенности. Например, зависимость L4 (это основной современный вариант интерпретации закона Мура) при рассмотрении роста показателей год за годом практически полностью соответствует ряду Фибоначчи. А зависимость L5 каждые 2 года и зависимость L1 каждые 10 лет дают коэффициент роста, близкий к значению числа π . Но самым главным является то, что абсолютное большинство процессов роста в современных компьютерных технологиях достаточно точно вписываются в один из 6-ти рассмотренных вариантов.

Третье обобщение

С учетом разнообразия процессов технического прогресса на протяжении всей истории цивилизации целесообразно на базе рассмотренного выше подхода получить дальнейшее обобщение закона Мура, позволяющее описывать весь спектр возможной динамики экспоненциальных процессов развития.

Для этого, во-первых, кроме базового периода роста в 20 лет, характерного для ИКР, следует рассматривать и более длительные базовые периоды, отличающиеся на один и более десятичных порядков от наиболее короткого базового периода.

Во-вторых, будем предполагать, что основной оценкой скорости технического прогресса во всех случаях будет количество десятичных порядков, на которое возрастает (или в некоторых случаях уменьшается) значение того или иного показателя в ходе развития техники и технологий. При этом соответствующую скорость роста целесообразно обозначить через S (начальный символ англоязычного слова speed, означающего скорость). Такое обозначение представляется целесообразным также и потому, что большинство процессов развития в технике могут быть описаны различными S -образными кривыми [12], начальная часть которых (примерно первая половина или несколько более) представляет из себя экспоненциальную или подобную ей кривую.

Для различных скоростей экспоненциального роста обозначение в целом будет иметь вид SK , где K – это численное обозначение скорости роста, равное количеству десятичных порядков роста за базовый период. Это, естественно, отнюдь не предполагает, что соответствующий процесс роста будет длиться на протяжении всего базового периода. Чаще всего рост носит экспоненциальный характер на протяжении всего нескольких десятилетий. Но, как показывает практика, для сравнительного анализа привязка к базовым периодам является наиболее целесообразной.

При рассмотрении технического прогресса можно ограничиться максимальным базовым периодом в 2 тысячи лет, т.к. за пределами данного периода процессы технического развития либо вообще плохо прослеживаются, либо просто недостаточно документированы для более-менее достоверных количественных оценок. В этом случае в обозначении K будет использоваться 3 десятичных знака. Нули справа в обозначении K в большинстве случаев могут отбрасываться.

В общем случае будем считать, что n – количество десятичных знаков (цифр), составляющих K . Тогда K показывает на сколько десятичных порядков вырастет соответствующее значение рассматриваемого параметра роста за $2 \cdot 10^n$ лет.

Символом S без цифровых коэффициентов будем обозначать переменную, принимающую значение K . Для того, чтобы значение S было корректным для различных n для определения его целесообразно использовать следующую формулу:

$$S = 10 \cdot 0, K.$$

Это, например, означает что для $S1$ $S=1$, для $S05$ $S=0,5$, для $S225$ $S=2,25$.

Минимальной единицей времени при расчетах будем считать один год, так в пределах года даже в случае самых быстрых темпов технического прогресса существенных изменений либо не наблюдается, либо они лежат в пределах возможных отклонений. Тогда для вычисления экспоненциально растущих значений P_i для любого i -го года ($i > 0$) при известном начальном значении P_0 может использоваться зависимость, аналогичная той, которая была приведена в работе [1] для обобщенного закона Мура:

$$P_i = P_0 \cdot 2^{S \cdot (Y_i - Y_0) / 6}, \text{ где}$$

Y_0 – начальный год действия соответствующей закономерности;

Y_i – текущий год действия соответствующей закономерности;

Использование такой формы зависимости, описывающей закономерности роста, позволяет обеспечить совместимость системы обозначений, используемой для первого и второго обобщений, с

обозначениями, описанными выше. При этом для шести вариантов закономерностей обобщенного закона Мура от $L1$ до $L6$ будем использовать, соответственно, обозначения от $S1$ до $S6$.

В таблице 2 приведены ежегодные коэффициенты роста sK для различных значений K , выраженных как X , $0X$ и $00X$ для 3-х вариантов базовых периодов.

Таблица 2 – Ежегодные коэффициенты роста для различных вариантов третьего обобщения закона Мура

| X | Коэффициенты роста за базовый период (20, 200, 2000 лет) | Ежегодные коэффициенты роста | | |
|----|--|------------------------------|--------------|----------------|
| | | 20 лет: sX | 200 лет: s0X | 2000 лет: s00X |
| 1 | 10 | 1,12 | 1,012 | 1,0012 |
| 2 | 100 | 1,26 | 1,023 | 1,0023 |
| 3 | 1 000 | 1,41 | 1,035 | 1,0035 |
| 4 | 10 000 | 1,58 | 1,047 | 1,0046 |
| 5 | 100 000 | 1,78 | 1,059 | 1,0058 |
| 6 | 1 000 000 | 2,00 | 1,072 | 1,0069 |
| 7 | 10 000 000 | 2,24 | 1,084 | 1,0081 |
| 8 | 100 000 000 | 2,51 | 1,096 | 1,0093 |
| 9 | 1 000 000 000 | 2,82 | 1,109 | 1,0104 |
| 10 | 10 000 000 000 | 3,16 | 1,122 | 1,0116 |

В общем случае при произвольных значениях K таблица 1 может использоваться для определения значения K в соответствии со следующим алгоритмом:

Шаг 1: Определяем, за какой в период Y (в годах) происходит рост в Z раз.

Шаг 2: Извлекаем корень степени Y из Z .

Шаг 3: В таблице 1 в колонке «20 лет» находим ближайшее значение, не превышающее полученное на шаге 2. Соответствующий номер строки X будет первым десятичным знаком значения K . В случае полного совпадения значений на данном шаге алгоритм завершается.

Шаг 4: Определяем, во сколько раз значение, полученное на шаге 2 отличается от значения, определенного на шаге 3.

Шаг 5: В таблице 1 в колонке «200 лет» находим ближайшее значение, не превышающее полученное на шаге 4. Соответствующий номер строки X будет вторым десятичным знаком значения K . В случае полного совпадения значений на данном шаге алгоритм завершается.

Шаг 6: Определяем, во сколько раз значение, полученное на шаге 4 отличается от значения, определенного на шаге 5.

Шаг 7: В таблице 1 в колонке «2000 лет» находим значение, ближайшее к полученному на шаге 6. Соответствующий номер строки X будет третьим десятичным знаком значения K . На этом алгоритм завершается.

Естественно, что альтернативой значениям S могут быть просто коэффициенты ежегодного

роста, но они. Как уже отмечалось, не дают достаточно наглядного представления о реальных многолетних темпах технического прогресса.

J-образные кривые развития

Четвертым, наиболее универсальным обобщением закона Мура можно считать описанное далее использование J-образных кривых развития.

Традиционно для описания динамики развития естественных и технических систем используются S-образные кривые (или просто S-кривые), впервые предложенные в XIX веке бельгийским математиком Пьером Ферхюльстом для моделирования динамики численности населения. Соответствующее уравнение было им названо логистическим (причина использования им именно такого названия остается невыясненной), в связи с чем соответствующие кривые также иногда определяются как логистические. К середине XX века S-кривые стали широко использоваться для описания динамики самых различных процессов развития, имеющих ресурсные ограничения. Уравнение кривой при этом имеет следующий вид:

$$\frac{dN}{dt} = rN \left(\frac{K - N}{K} \right),$$

где

N – некоторое отслеживаемое значение, изменяющееся в соответствии с логистической кривой;

K – предельное значение для N ;

r – масштабный коэффициент.

На начальном этапе развития по S-траектории, когда текущее значение соответствующего критерия развития намного меньше предельного, кривая развития может рассматриваться как экспоненциальная, уравнение которой выглядит следующим образом:

$$\frac{dN}{dt} = rN.$$

Экспоненциальные траектории развития принято обозначать как J-кривые. Именно этот тип кривых развития, как будет, показано далее, является наиболее типичным для технических систем. При этом в большинстве случаев экспоненциальные процессы не переходят в классические S-кривые, как это имеет место в большинстве естественных и социально-экономических систем, а образуют специфические каскады J-кривых 2-х основных типов:

JJ-кривые, когда развитие доходит до некоторого предельного для текущей J-кривой значения и далее развитие идет в соответствии с новой J-кривой, как правило, существенной более пологой, чем предыдущая;

jJ-кривые, когда при достижении предельного для текущей J-кривой значения дальнейшее развитие идет в соответствии с новой более пологой J-кривой, начинающейся не с предельного значения, а с некоторого промежуточного, оптимального для дальнейшего развития, значения.

Выбор обозначения S как показателя темпов роста в третьем обобщении закона Мура при этом был обусловлен не только тем, что это начальный символ англоязычного слова speed, означающего скорость, но и тем, что большинство процессов развития в различных системах могут быть описаны различными S-образными кривыми, начальная часть которых представляет собой практически экспоненциальную или квазиэкспоненциальную кривую. В случае JJ-кривых и различных их каскадов в виде jJ-кривых и вместо S целесообразно использовать значение J или J_p , где p – это длительность базового периода в годах. Для десятилетнего базового периода будет, соответственно, использоваться значение J_{10} , для двадцатилетнего J_{20} и т.д., что позволит в общем случае при необходимости вводить и использовать базовый период произвольной длительности. По умолчанию в качестве основного будет предполагаться базовый период в 20 лет, для обозначения которого будет использоваться J без каких-либо индексов, что, соответственно, будет эквивалентно использованию обозначения J_{20} .

Для упрощения вычисления J_p целесообразно вывести единую формулу, исходными значениями которой будут начальные и конечные годы и значения соответствующего периода экспоненциального роста, и реализовать ее вычисление с помощью, например, такого инструмента как MS Excel. Результатом вычисления будет число порядков роста за базовый период, имеющее в общем случае целую часть, эквивалентную для J_{20} значению первой цифры при обозначении S, а цифры дробной части будут эквивалентны соответственно второй, третьей и т.д. цифрам при обозначении S. Например, значение $J=0,05$ будет соответствовать S005, а значение $J=1,23$ будет соответствовать S123. Это означает, в частности, что вычисление J автоматически позволяет получить с требуемой точностью все цифры для обозначения S. В общем случае для сравнительного анализа достаточно просто вычислять и указывать для соответствующих J-кривых значение J_p .

При экспоненциальном развитии коэффициент ежегодного роста R_u при известных начальных и конечных значениях экспоненциально изменяющейся величины Z для

периода от Yb до Ye вычисляется следующим образом:

$$Ry = \sqrt{(Ye-Yb) \left(\frac{Ze}{Zb}\right)},$$

где

Yb – начальный год;

Ye – конечный год;

Zb - начальное значение;

Ze – конечное значение.

Рост на порядок за базовый период p (в годах) обеспечивается при ежегодном коэффициенте роста Rp , который вычисляется следующим образом:

$$Rp = \sqrt[p]{10}.$$

На базе данных двух значений вычисляется искомый показатель Jp – количество порядков, на которое происходит рост за базовый период:

$$Jp = \log_{Rp} Ry.$$

Для реализации вычисления Jp на базе MS Excel используется функция $LOG(X;a)$, которая возвращает логарифм заданного числа X по заданному основанию a . При этом в качестве X используется значение Ry , а в качестве a – значение Rp .

Таким образом, для произвольной многолетней экспоненциальной зависимости, для которой известны значения в некоторый начальный и некоторый конечный год, можно определить показатель Jp , определяющий скорость роста за базовый период, что можно считать максимальным обобщением закона Мура.

Более детальный анализ различных зависимостей на примере множества конкретных примеров, связанных с развитием как компьютерных систем и технологий, так и прочих технических систем, приведен в работах [13-24], которые и послужили основой при написании данной статьи.

Выводы

На базе анализа различных вариантов закона Мура и связанных с ним прочих эмпирических закономерностей предложена четыре обобщенных варианта единого описания подобного рода законов и закономерностей, отличающиеся степенью детализации и универсальности. Предполагается также, что выявленная система закономерностей в области развития компьютерных систем и технологий может быть в той или иной степени обобщена применительно к другим техническим системам и технологиям, которые также (хотя и существенно более медленно) развиваются экспоненциально.

Литература

1. Мурки Т. Закон Мура против нанометров. // iXBT.com: Сайт о высоких технологиях, 2011. – Электр. ресурс. URL: <http://www.ixbt.com/cpu/microelectronics.shtml> (10.07.14).
2. Мэлоун М. The Intel: как Роберт Нойс, Гордон Мур и Энди Гроув создали самую влиятельную компанию в мире. – М.: Эксмо, 2015. – 680 с.
3. Moore G. E. Cramming more components onto integrated circuits / Electronics, vol. 38, no. 8, Apr. 1965. P. 114–117.
4. Половинкин А.И. Основы инженерного творчества. – М.: Машиностроение, 1988. – 368 с.
5. Громов Г. Р. Национальные информационные ресурсы: проблемы промышленной эксплуатации. – М.: Наука, 1984. – 240 с.
6. Громов Г. Р. Очерки информационной технологии. – М.: Инфоарт, 1992. – 334 с.
7. Moore G. Progress in digital integrated electronics / Proc. of the International Electron Devices Meeting (IEDM'75), vol. 21, 1975. P. 11-13.
8. Moore G. Lithography and the Future of Moore's Law // SPIE, Vol. 2438, 1995. P. 2-17.
9. Rupp K. and Selberherr S. The Economic Limit to Moore's Law // IEEE Transactions on Semiconductor Manufacturing Journal, vol. 24, no. 1, February 2011. – 4 p.
10. Feng W. The Importance of Being Low Power in High Performance Computing // Cyberinfrastructure Technology Watch Quarterly, vol. 1, no. 3, August 2005. P. 12-20.
11. Аноприенко А.Я. Системодинамика ноотехносферы: основные закономерности / Системный анализ в науках о природе и обществе. – 2014. – №1(6)-2(7). – С. 11-29.
12. Мартино Дж. Технологическое прогнозирование. – М.: Прогресс, 1977. – 592 с.
13. Аноприенко А.Я. Пределы информатики // «Информация и рынок». Теоретический и научно-практический журнал. – 1993. – №2-3. С. 10-14.
14. Аноприенко А.Я. Компьютерные науки и технологии: следующие 50 лет // Материалы научно-технической конференции «Информационные управляющие системы и компьютерный мониторинг (ИУС и КМ 2011)» – 12-13 апреля 2011 г., Донецк, ДонНТУ, 2011. Т.1. С. 7-22.
15. Аноприенко А.Я. Компьютерные науки и технологии в прошлом, настоящем и будущем // Материалы V международной научно-технической конференции «Информатика и компьютерные технологии» – 24-26 ноября

- 2009 г., Донецк, ДонНТУ, 2009. С.15-26.
16. Анопrienко А.Я. Модели эволюции компьютерных систем и средств компьютерного моделирования // Материалы пятой международной научно-технической конференции «Моделирование и компьютерная графика» 24-27 сентября 2013 года, Донецк, ДонНТУ, 2013. С. 403-423.
 17. Анопrienко А.Я. Основные закономерности эволюции компьютерных систем и сетей // Научные труды ДонНТУ. Серия «Проблемы моделирования и автоматизации проектирования». Выпуск № 1 (12) – 2 (13): Донецк: ДонНТУ, — 2013. С. 10–32.
 18. Анопrienко А.Я. Закономерности развития компьютерных систем // «Научная дискуссия: инновации в современном мире». №10 (18): Сборник статей по материалам XVIII международной заочной научно-практической конференции. – М.: Изд. «Международный центр науки и образования», 2013. – С. 19-29.
 19. Анопrienко А.Я. Система закономерностей развития средств и методов компьютеринга // Материалы V всеукраинской научно-технической конференции «Информационные управляющие системы и компьютерный мониторинг (ИУС и КМ 2014)» – 22-23 апреля 2014 г., Донецк, ДонНТУ, 2014. В 2-х томах. Т. 1. С. 11-23.
 20. Анопrienко О.Я., Варзар Р.Л., Иваница С.В. Закономерности развития аналого-цифровых преобразователей и перспективы использования постбинарного кодирования // Научные труды ДонНТУ. Серия: «Информатика, кибернетика и вычислительная техника». Выпуск 1 (19). – Донецк: ДонНТУ, 2014. С. 5-10.
 21. Анопrienко А.Я. Периодическая система развития компьютерных систем и перспективы нанокompьютеризации // Инновационные перспективы Донбасса: Материалы международной научно-практической конференции. Донецк, 20-22 мая 2015 г. Том 5. Компьютерные науки и технологии. – Донецк: Донецкий национальный технический университет, 2015. С. 14-22.
 22. Анопrienко А.Я. Программная инженерия и обобщенный закон Мура // Первая международная научно-практическая конференция «Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2016)». Донецк, 16-17 ноября 2016 г. Сборник научных трудов. – Донецк: ГОУ ВПО «Донецкий национальный технический университет», 2016. – С. 41-47.
 23. Анопrienко А.Я. Системодинамика техносферы: как измерить технический прогресс // Системный анализ и информационные технологии в науках о природе и обществе, 2015. № 1(8)-2(9). С. 47-58.
 24. Анопrienко А.Я. Закономерности развития компьютерных технологий и обобщенный закон Мура // Вестник Донецкого национального технического университета, №2 (2), 2016. С. 3-17.

Anopriyenko A. Generalizations of Moore's Law. Empirical observations, which characterize the dynamics of the exponential development of computer technology, are systematized and generalized. It is shown that the variety of basic regularities, including those characterizing Moore's law, can be represented as a series of generalized relatively simple dependencies. It is assumed that the revealed system of regularities has sufficient predictive properties of necessary accuracy for the short and long term.

Статья поступила в редакцию 7.09.2017
Рекомендована к публикации доктором технических наук В.Н. Павлышом

УДК 004.023

Подсистема оптимизации работы гибких производственных систем с использованием многокритериальных генетических алгоритмов

Бабич К. К., Секирин А.И., Новиков Д.Д.
Донецкий национальный технический университет г. Донецк
Кафедра автоматизированных систем управления
E-mail: bkristyaa@gmail.com

Бабич К. К., Секирин А.И., Новиков Д. Д. Подсистема оптимизации работы гибких производственных систем с использованием многокритериальных генетических алгоритмов. В статье обоснована актуальность проблемы, проведен краткий обзор основных алгоритмов и методов моделирования, используемых для разрешения схожих проблем, описана математическая постановка задачи и предложенное решение.

Ключевые слова: гибкие производственные системы, максимизация среднего коэффициента загрузки, генетический алгоритм, модель

Актуальность проблемы

В практике человеческой деятельности, будь то профессиональная сфера или повседневная жизнь, постоянно возникают задачи выбора, предполагающие в результате принятие решения. Только в ряде случаев, лицо принимающее решение осуществляет выбор (принимает решение) интуитивно, опираясь на собственный опыт и здравый смысл, а решение более сложных задач требует особого подхода, так как в данном случае задача принятия решения представляет собой, по сути, уже оптимизационную задачу.

Простейшая ситуация выбора решений соответствует случаю, когда преследуется единственная цель, и эта цель может быть формально задана в виде скалярной функции критерия качества выбора или значения критерия качества могут быть получены для любого допустимого набора значений аргументов. Предполагается также, что известна область определения параметров, входящих в целевую функцию, накладываемые на них ограничения, или для любой заданной точки может быть установлено, является ли она допустимым выбором. В такой ситуации задача выбора решения может быть формализована и описана моделью математического программирования.

Методы решения задач математического программирования с одним критерием

интенсивно разрабатывались последние 40 лет. Изучение таких методов, однако, отражало самый ранний и простой этап в развитии математического

программирования. По мере того как мы стремительно вступаем в век информатики, становится ясно, что практически любая серьезная задача характеризуется более чем одним критерием. Ощущается необходимость оценивать альтернативные решения с точки зрения нескольких критериев.

В процессе управления сложными технологическими комплексами и организационно-техническими системами необходимо постоянно принимать непростые решения, связанные с учетом многих критериев протекания производственных процессов и ограничений на ресурсы. Без использования возможностей современной вычислительной техники достаточно сложно сделать оптимальный выбор. В этой связи необходимо разрабатывать и внедрять системы поддержки принятия решений.

Необходимость внедрения информационных систем управления более остро стоит перед предприятиями машиностроительной отрасли. Машиностроение является одной из наиболее наукоемких, сложных, требующих высококвалифицированных

высококвалифицированных кадров отраслей материального производства. Уровень развития машиностроения определяет уровень развития государства в целом.

Основой для создания компьютеризированных производственных систем являются гибкие производственные системы (ГПС). Это технологические структуры с законченным производственным циклом (участок, цех) и различной степенью автоматизации. ГПС включает в себя автоматизированное технологическое оборудование для механообработки деталей,

а также вспомогательное оборудование: автоматизированный транспорт и автоматизированную складскую систему.

ГПС представляет собой сложный динамический объект. Для эффективного использования такого объекта необходимо формировать и динамически корректировать с учетом изменения ситуации на производстве оптимальное расписание работы оборудования. На реальный производственный процесс влияет ряд внешних факторов:

- сбои в транспортно-складской системе;
- введение в производство новой детали;
- выход из строя технологического оборудования.

Для разработки метода оптимизации необходима модель ГПС, которая обеспечит близкое к реальности отображение процесса работы оборудования и движения материальных и информационных потоков.

Фактически задача оптимизации работы состоит из двух этапов:

- разработка модели ГПС;
- разработка алгоритма оптимизации.

Классическая схема решения задач производственной оптимизации представлена на рисунке:

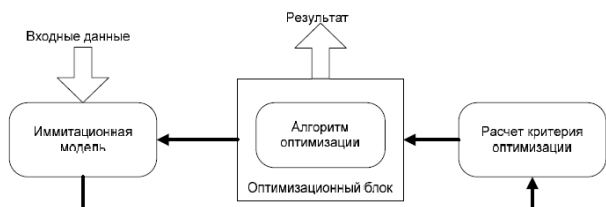


Рисунок 1. Схема подсистемы оптимизации

Применяемые методы, модели и алгоритмы для решения задачи

Математическое моделирование – это средство изучения реального объекта, процесса или системы путем их замены математической моделью, более удобной для экспериментального исследования с помощью ЭВМ.

Модель является приближенным представлением реальных объектов, процессов или систем, выраженным в математических терминах и сохраняющим существенные черты оригиналов. Математические модели описывают основные свойства объекта, его параметры, внутренние и внешние связи.

Для моделирования работы гибких производственных систем чаще всего

используются следующие методы моделирования:

- аналитические модели;
- модели, построенные на теории сетей и графов;
- имитационные модели;
- объектно-ориентированный подход.

Проведенный сравнительный анализ моделей, применяемых для формализации работы ГПС показал, что на сегодня наиболее эффективными являются подходы, основанные на использовании сетей и графов или имитационное моделирование. Однако указанные модели

не лишены недостатков, главным из которых является формализация с допущениями с целью упрощения модели, что негативно влияет на оперативность и оптимальность управления, а также на эффективность функционирования ГПС в целом.

Для оптимизации работы ГПС применяются следующие алгоритмы:

- метод линейного программирования;
- метод динамического программирования;
- метод ветвей и границ;
- генетические алгоритмы.

При использовании первых трех алгоритмов оптимальное расписание работы технологического комплекса либо вообще не может быть получено, либо не может быть получено в приемлемые сроки. Решения получены лишь для простейших случаев. Выходом из ситуации является использование эвристических методов поиска, таких как генетические алгоритмы. Механизм развития и наследования позволяет последовательно улучшать хромосомы в каждой новой популяции, что приводит к получению субоптимальных (близких к оптимальным) решений.

Математическая постановка

В общем виде задачу оптимизации можно описать следующим образом: имеется n деталей, которые необходимо обработать на L станках. При этом необходимо составить расписание работы оборудования, очередность запуска деталей в производство и размеры партий запуска, которые будут удовлетворять одновременно нескольким критериям оценки качества расписания.

Запишем основные критерии, по которым может быть эффективности полученного расписания:

1. Минимизация длительности цикла изготовления деталей

$$T_{\text{опт}} = T \rightarrow \min \quad (1)$$

где T - общее время производственного цикла.

$$T = \sum_{k=1}^n \left(\sum_{j=1}^n T_{ij} + \sum_{j=1}^n \alpha_{ij} \right) \quad 2)$$

где T_{ij} - длительность цикла технологической операции i -ой партии деталей, α_{ij} - длительность простоя перед началом выполнения j -ой технологической операции над i -ой партией деталей;

2. Максимизация среднего коэффициента загрузки технологического оборудования (максимизация загрузки оборудования)

$$K_{\text{оптз}} = \frac{1}{m} * \sum K_3 \rightarrow \max, \text{ или } K_{\text{оптз}} = \frac{1}{m} * \sum K_3 \rightarrow \min \quad 3)$$

где m - количество оборудования, K_3 - коэффициент загрузки i -го оборудования.

Коэффициент загрузки i -го оборудования определяется отношением времени производительной работы к общему времени по формуле:

$$K_3 = \frac{\sum_{j=1}^n T_{ij}}{\sum_{j=1}^n T_{ij} + \sum_{j=1}^n \alpha_{ij}} \quad 4)$$

3. Минимизация временных затрат на перенастройку оборудования

$$t_H = \sum_{i=1}^m \sum_{j=1}^n \tau_{ij} \rightarrow \min \quad 5)$$

где t_H - общее время, затраченное на перенастройку оборудования, τ_{ij} - время, затраченное на перенастройку i -го станка j -ый раз, n - количество перенастроек

Т.е. фактически необходимо минимизировать количество перенастроек:

$$n \rightarrow \min \quad 6)$$

При этом необходимо учитывать следующие ограничения:

1. Ограничения по объемам изготовления:

$$N_{\text{пл}} = N_{\Phi} \quad 7)$$

где N_{Φ} - фактическое изготовленное количество деталей i -го типа ($i=1 \dots K$), $N_{\text{пл}}$ - заданное в производственной программе количество деталей i -го типа, K - номенклатура изготавливаемых деталей.

2. Ограничения по срокам изготовления:

$$T_{\text{пл}} \geq T_{\Phi} \quad 8)$$

где T_{Φ} - фактический срок изготовления i -ой детали ($i=1 \dots K$), $T_{\text{пл}}$ - директивный срок изготовления i -ой детали.

3. Ограничение по фонду времени работы технологического оборудования:

$$\sum_{i=1}^n N_{\text{пл}} \sum_{j=1}^m T_{ij} \leq R_i \quad 9)$$

где T_{ij} - длительность выполнения технологической операции, R_i - ресурс i -ой группы оборудования.

В реальных условиях производства на функционирование цеха механообработки оказывают влияние различные возмущающие воздействия $\{V_k\}$, которые приводят к отклонениям выпуска деталей от производственной программы.

Задачей управления является обеспечение выпуска продукции согласно производственной программе ($N_{\text{пл}}$) по количеству и в установленные сроки при эффективном использовании ресурсов (R_i) в условиях действия возмущений (V_k). Обеспечение высокой эффективности использования ресурсов (R_i) и функционирования цеха в целом достигается за счет оптимизации расписаний работы оборудования.

Процесс изготовления деталей D_i ($i=1 \dots n$) разбивают на технологические операции O_{ij} ($i=1 \dots n$; $j=1 \dots m$). Детали одного типа объединяются в транспортные партии и в разрезе расписаний рассматриваются как технологическая единица. Каждая операция может быть выражена:

$$O_{ij} = \langle H_{ij}, T_{ij} \rangle \quad 10)$$

где H_{ij} - номер группы технологического оборудования; T_{ij} - продолжительность выполнения операции.

Технологический маршрут представляет собой последовательность выполняемых операций, которые проходят i -я деталь в процессе обработки:

$$M_i = \langle O_{i1}, O_{i2} \dots O_{im} \rangle \quad 11)$$

Операция O_{ij} должна выполняться без перерыва с самого начала. Если обозначить через t_{ij} - время начала выполнения операции O_{ij} , а через t'_{ij} - момент окончания обработки, то должно выполняться равенство:

$$t'_{ij} = t_{ij} + T_{ij} \quad 12)$$

Очевидно, что время начала обработки зависит от времени выполнения предыдущих операций и всегда выполняется неравенство: $t_{ij} \leq t_{ij+1}$. Тогда совокупность $\{t_{ij}\}$ ($i = 1 \dots n$; $j = 1 \dots m$), удовлетворяющих всем технологическим и

временным ограничениям, является расписанием работы.

Предлагаемое решение

Для построения адекватной модели технологического комплекса используется объектно-ориентированный подход. Учитывая модульную структуру ГПС выделены следующие классы:

- гибкий производственный модуль (ГПМ);
- автоматизированный транспортный модуль (АТМ);
- автоматизированный склад (АС);
- система управления (СУ).

Для управления комплексом и создания событий введен дополнительный управляющий класс «диспетчер событий».

В качестве алгоритма оптимизации используется двухуровневый генетический алгоритм.

$$C_i^1 = [m_1, m_2 \dots m_n], C_i^2 = [k_1, k_2 \dots k_n] \quad 13)$$

где m и k – натуральные числа, n – размер популяции

Первый уровень хромосом кодирует различные варианты последовательностей запуска деталей по их типам на технологический участок. Каждой хромосоме первого уровня сопоставляется хромосома второго уровня, в которой содержится информация о величине партий запуска для каждого типа детали. Начальные решения (хромосомы первой популяции) формируются случайным образом, затем генетический алгоритм производит перебор очередности и размеров партий запуска, поступающих в обработку.

Для реализации многокритериальной оптимизации используется метод взвешенной функции, он является естественным развитием классических методов оптимизации, где новая «общая» целевая функция строится из заданных в виде взвешенной суммы:

$$F(x) = \sum_{i=1}^k w_i f_i(x), \text{ где веса } w_i \in [0,1] \text{ и } \sum_{i=1}^k w_i = 1 \quad 14)$$

Здесь каждой целевой функции $f_i(x)$ присваивается свой вес w_i и задача сводится к скалярному случаю. При этом различные веса w_i дают разные решения в смысле Парето.

Программная реализация

Подсистема состоит из двух программных модулей:

- Блок моделирования;
- Блок оптимизации.

Для реализации используется среда MS Visual Studio 2010, в качестве языка программирования выбран C#.

Результаты работы

Моделирование работы гибких производственных систем механообработки позволяет получать оценки загрузки всех видов оборудования и выявлять «узкие места», снижающие эффективность работы ГПС. Проведение на модели анализа различных вариантов их устранения позволяет разработать практические рекомендации, обеспечивающие повышение загрузки технологического оборудования. Применение модифицированного двухуровневого генетического алгоритма совместно с объектной моделью позволяет получать субоптимальные расписания работы оборудования по нескольким критериям, что обеспечивает высокую эффективность функционирования ГПС механообработки.

Выводы

Разработанная объектная модель ГПС позволила получить и проанализировать загрузку различных видов оборудования: гибких производственных модулей, автоматизированного транспорта и склада. Применение генетических алгоритмов позволяет получать близкие к оптимальным решения в допустимые сроки. Проведенные исследования

показали целесообразность и эффективность применения данного подхода в управлении автоматизированными технологическими комплексами.

Список литературы

1. Секирин А.И. Использование генетических алгоритмов для оптимизации работы автоматизированных технологических комплексов механообработки [электронный ресурс]. Режим доступа: <http://masters.donntu.edu.ua/2009/kita/aleksandrova/library/article10.htm>
2. Оптимизация моделей процессов производства [электронный ресурс]. Режим доступа: http://revolution.allbest.ru/manufacture/0003752_2_0.html
3. Сачко Н.С. Организация и оперативное управление

машиностроительным производством / Н.С. Сачко – Новое знание, 2006. – 520 с.

4. Шеховцов А.В., Крючковский В.В., Мельник А.Н. «Решение многокритериальной оптимизации с использованием адаптивных алгоритмов» [электронный ресурс]. Режим доступа: <http://aaecs.org/shehovcov-av-kryuchkovskii-vv-melnik-a-n-reshenie->

[mnogokriterialnoi-optimizacii-s-ispolzovaniem-adaptivnih-algoritmov.html](http://mat.1september.ru/mnogokriterialnoi-optimizacii-s-ispolzovaniem-adaptivnih-algoritmov.html)

М. Скворцова Математическое моделирование [электронный ресурс]. Режим доступа: <http://mat.1september>

Babich K., Sekirin A., Novikov D. Subsystem for optimizing the operation of flexible production systems using multicriteria genetic algorithms. The article substantiates the urgency of the problem, provides a brief overview of the main algorithms and modeling methods used to solve similar problems, describes the mathematical formulation of the problem and the proposed solution.

Keywords: Flexible production systems, maximizing of the average load factor, genetic algorithm, model

Статья поступила в редакцию 20.09.2017

Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 004.7

Вычислительные структуры для размещения файлов по узлам компьютерной сети

Бельков Д.В.

Донецкий национальный технический университет

belkov65@list.ru

Бельков Д.В. Вычислительные структуры для размещения файлов по узлам компьютерной сети. Статья предлагает метод и вычислительные структуры для оптимизации размещения файлов в компьютерных сетях. Для повышения эффективности компьютерных сетей за счет рационального размещения файлов сформулирована задача рационального размещения файлов. Для ускорения решения задачи предложены вычислительные структуры, ориентированные на ее решение.

Ключевые слова: Файлы, узлы компьютерной сети, специализированные процессоры для размещения файлов.

Введение

Быстрое развитие информационных и коммуникационных технологий в современном мире привело к широкому распространению распределенных систем обработки данных на основе компьютерных сетей. Одним из способов повышения эффективности функционирования компьютерных сетей является оптимизация размещения файлов. Поэтому задача оптимального размещения файлов по узлам сети имеет важное практическое значение. Эта задача относится к классу NP - трудных. Точные методы можно применять только для решения задачи малой размерности. В случае большой размерности необходимо использовать приближенные методы.

В случае большого числа файлов программная реализация метода их распределения на однопроцессорной ЭВМ становится неэффективной, т.к. время распределения пропорционально числу файлов. По этой причине для оптимизации размещения большого числа файлов необходима разработка специализированных процессоров, позволяющих сети было минимальным.

В данной работе для уменьшения времени отклика оптимизируется интенсивность трафика. При функционировании сети в каждом узле образуются 2 типа запросов: сетевой запрос, для обработки которого необходим файл, не содержащийся в том узле, где возник запрос и локальный запрос, для обработки которого необходим файл, содержащийся в том узле, где возник запрос. Критерием оптимальности размещения файлов является суммарный поток локальных запросов, инициированных в узлах в

ускорить распределение файлов за счет параллельных вычислений.

Таким образом, необходимость повышения эффективности функционирования компьютерных сетей делает актуальной разработку методов и вычислительных структур для размещения файлов в компьютерных сетях.

Цель статьи - повышение эффективности работы компьютерной сети за счет оптимизации размещения файлов с применением специализированных вычислительных структур. Задача работы - разработать вычислительные структуры для размещения файлов в компьютерной сети, позволяющие ускорить размещение большого количества файлов.

Задача размещения файлов

Для статического размещения файлов по узлам компьютерной сети необходимо при фиксированных значениях интенсивностей запросов к файлам так распределить файлы по узлам компьютерной сети, чтобы время отклика единицу времени. Чем больше суммарный поток локальных запросов, тем меньше время отклика сети.

Обозначим: F_{ij} - количество запросов к файлу i из узла j в единицу времени; $X_{ij} = 1$, если файл i расположен в узле j , иначе $X_{ij} = 0$; V_i - объем файла i ; C_i - количество копий файла i ; B_j - объем узла j , $i=1...m$, $j=1...n$.

Для рационального использования памяти узлов целесообразно минимизировать

объем их свободной памяти. Поэтому коэффициент заполнения узлов, равный отношению V_i / B_j необходимо максимизировать.

Задача размещения файлов по узлам компьютерной сети имеет вид:

Целевая функция

$$L = \sum_{i=1}^m \sum_{j=1}^n F_{ij} V_i X_{ij} / B_j = \sum_{i=1}^m \sum_{j=1}^n L_{ij} X_{ij} \rightarrow \max \quad (1)$$

Ограничения:

$$X_{ij} \in \{0,1\}, \sum_{j=1}^n X_{ij} = C_i \quad i=1...m \quad (2)$$

$$\sum_{i=1}^m V_i X_{ij} \leq B_j \quad (3)$$

В задаче (1)-(3) необходимо найти матрицу размещений файлов X . Максимизируется суммарный поток локальных запросов. Это приводит к повышению эффективности функционирования сети за счет уменьшения среднего времени отклика на запросы к файлам.

Метод решения задачи (1)-(3), предложенный в [1], состоит из $C_i \cdot m$ этапов, C_i - количество копий файла i . На этапе i выполняется процедура размещения файла i в один из n узлов. Процедура состоит из 3 шагов. На первом шаге сравниваются свободный объем каждого узла и объем файла i чтобы найти те узлы, в которые файл помещается по размеру. На втором шаге, среди найденных узлов определяется узел с наибольшим значением $L_{ij} = F_{ij} V_i / B_j$, $j=1, \dots, n$. На третьем шаге файл размещается в этот узел. Исходными данными для процедуры служат значения F_{ij}, V_i, C_i, B_j .

Величина B_j первоначально совпадает со свободным объемом узла j - U_j . В результате работы процедуры формируются новые значения U_j и строка i матрицы X . Временная сложность

метода составляет $O(n \sum_{i=1}^m C_i)$, где m - количество файлов, n - число узлов, C_i - число копий файла i .

Сетевые процессоры

Сетевые процессоры являются специализированными программируемыми устройствами для обработки сетевого трафика с высокой скоростью. Сетевые процессоры применяются в сетях ATM на канальном, сетевом и транспортном уровнях стека протоколов OSI.

На сетевом уровне используется протокол IP. Единица информации, передаваемая при помощи IP, называется IP-дейтаграммой. Важной функцией IP является маршрутизация дейтаграмм. Это протокол без установки соединения, который не гарантирует сквозную передачу данных, т.е. дейтаграммы могут теряться. Две дейтаграммы, последовательно отправленные в одно и то же место назначения, маршрутизируются независимо, могут следовать по разным маршрутам и прибыть в место назначения в порядке, противоположном отправке.

Протокол UDP является простым посредником между сетевым и прикладным уровнями. Он не выполняет никаких функций по обеспечению надежности передачи, не устанавливает логического соединения, не нумерует и не упорядочивает пакеты данных. Задача транспортного уровня заключается в передаче данных между любыми прикладными процессами, выполняющимися на любых узлах сети.

За годы использования стек TCP/IP накопил большое количество протоколов и сервисов прикладного уровня. К ним относятся протокол пересылки файлов FTP, протокол эмуляции терминала TELNET, гипертекстовые сервисы доступа к удаленной информации (HTTP), сетевая файловая система (NFS), протокол SNMP для организации сетевого управления. Структура сетевого процессора, предложенная в работе [2], показана на рисунке 1.

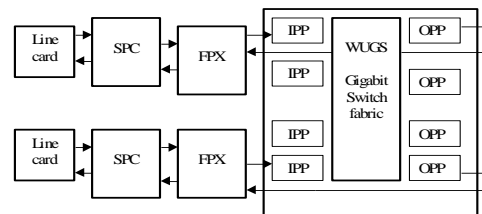


Рисунок 1 – Структура сетевого процессора

Центральным компонентом сетевого процессора является полнофункциональный ATM - переключатель (ATM-switch) WUGS. С ним соединены два дополнительных блока: smart port card (SPC) и field programmable port extender (FPX).

Блок SPC необходим для сложной обработки пакетов.

Блок FPX - это перепрограммируемая логика для пользовательских приложений. Структура блока показана на рисунке 2.

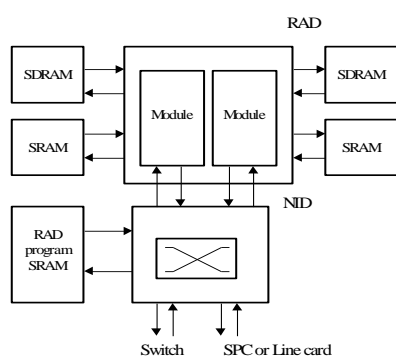


Рисунок 2 – Структура FPX

Блок содержит сетевой интерфейс (NID), реализованный на ПЛИС Xilinx XCV600E, и перепрограммируемое устройство для прикладных приложений (RAD), реализованное на ПЛИС Xilinx XCV2000E. ПЛИС NID служит для соединения RAD с WUGS, а также имеет логику для динамического перепрограммирования RAD. ПЛИС RAD содержит два программируемых пользователем модуля, которые соединены с блоками SRAM и SDRAM [4].

На рисунке 3 показана концепция сетевых процессоров (network wrapper concept), предложенная в работе [3].

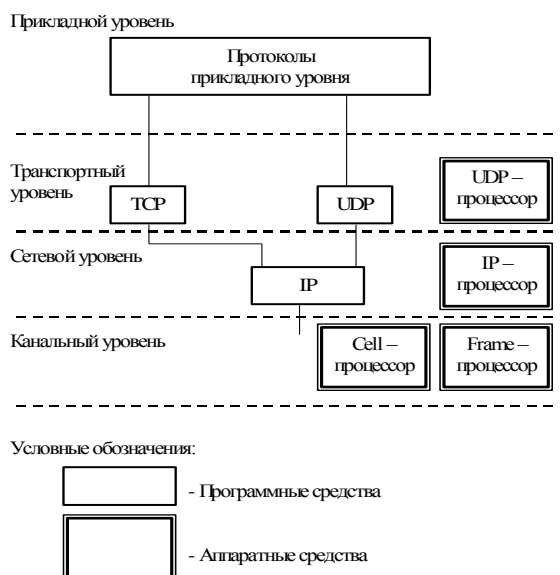


Рисунок 3 - Концепция сетевых процессоров

Cell-процессор получает пакеты (cells), обрабатывает ATM – заголовки и распределяет пакеты по виртуальным каналам. В сетях ATM пакеты имеют фиксированный размер. Для использования данных произвольного размера фрейм – процессор организует пакеты в кадры [4].

IP-процессор необходим для поддержки протокола IP. Маршрутизатор, построенный на основе IP – процессора, поддерживает таблицу маршрутизации, которая применяется для нахождения следующего маршрутизатора или хоста. UDP-процессор необходим для поддержки протокола UDP.

В концепции сетевых процессоров, предложенной в работе [4], прикладной уровень не рассматривается. Однако на прикладном уровне можно повысить эффективность функционирования сети за счет оптимизации времени отклика. Оптимизация должна осуществляться в режиме реального времени, поэтому требуется ее аппаратная поддержка. В статье предлагается расширить концепцию сетевых процессоров на прикладной уровень за счет использования спецпроцессора для оптимизации размещения файлов. Он должен быть ориентирован на решение задачи (1)-(3).

Спецпроцессор для размещения файлов

Задачу оптимизации размещения файлов по узлам вычислительной сети нужно решать в режиме реального времени. Поэтому необходима аппаратная реализация метода ее решения. На рисунке 4 показана структура вычислительной системы, которая состоит из спецпроцессора, буферной памяти и основного процессора. Буферная память имеет область входных регистров и выходной регистр. Система последовательно выполняет m этапов. Каждый этап имеет 5 шагов. На первом шаге этапа i основной процессор пересылает исходные данные для формирования строки i матрицы X во входные регистры. На втором шаге спецпроцессор считывает эти данные. На третьем шаге спецпроцессор структурно выполняет процедуру размещения файла i в узел. На четвертом шаге спецпроцессор передает результаты работы процедуры в выходной регистр. На пятом шаге основной процессор считывает эти результаты из буферной памяти.



Рисунок 4 – Структура вычислительной системы

Входные данные являются r – разрядными целыми числами без знака. Выходные данные – строка i матрицы X , $X_{ij} \in \{0;1\}$. Входные данные размещены в $2(n+1)$ регистрах разрядностью r , выходные данные размещены в регистре разрядностью n . На первом шаге метода $U_j = B_j, j=1,2,\dots,n$. На четвертом шаге метода обновляются значения

U_j во входных регистрах, и заполняется выходной регистр.

Структура спецпроцессора для размещения файлов показана на рисунке 5. В отличие от структуры, приведенной в работах [5,6], она позволяет решать задачу, в которой файлы могут иметь несколько копий. Описание модулей структуры дано в следующем разделе.

Каждый модуль построен на основе специализированного однородного процессора. Такой процессор является итеративной сетью, ориентированной на выполнение заданной базовой операции. Итеративная сеть есть логическая сеть, состоящая из одинаковых и одинаково связанных между собой элементов (ячеек). Особенностью итеративных сетей является то, что в них осуществляется прямое отображение алгоритмов в схемы. За счет прямого моделирования алгоритмов, итеративные сети позволяют синтезировать безызбыточные цифровые устройства высокой производительности. Специализированные однородные процессоры реализуют базовые операции структурно за один такт и относятся к компьютерам типа SIMD с крупноблочными базовыми операциями [7,8]. Процессоры обладают регулярной структурой и удобны для реализации в базе ПЛИС

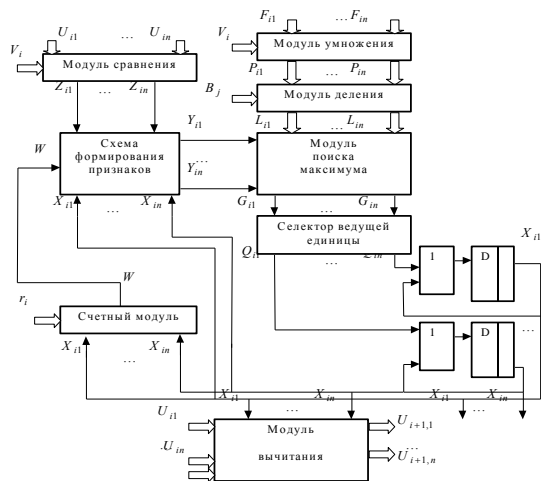


Рисунок 5 – Структура спецпроцессора

Процесс формирования элементов строки X_i происходит следующим образом. На первом шаге значения X_{ij} равны нулю, $Z_{ij} = 1$, если $U_{ij} \geq V_i$, иначе $Z_{ij} = 0$. На втором шаге осуществляется поиск максимального значения среди значений L_{ij} , таких, что $Z_{ij} = 1$. В результате этого, одно из значений X_{ij} устанавливается в единицу.

Модули спецпроцессора

Модуль сравнения формирует элементы строки Z_i : $Z_{ij} = 1$, если файл i может поместиться в узел j , иначе $Z_{ij} = 0$. Модуль имеет локальную входную и выходную память, операционное устройство и устройство управления. Входная память модуля сравнения состоит из g – разрядных регистров, хранящих числа U_j и V_i , $j=1,2,\dots,n$. Выходная память представляет собой n -разрядный регистр элементов вектора Z_i : $Z_{i1}, Z_{i2}, \dots, Z_{in}$. Операционное устройство, показанное на рисунке 6, состоит из n блоков, работающих параллельно. Блок j , показанный на рисунке 7, формирует элемент Z_{ij} : $Z_{ij} = 1$, если $U_{ij} \geq V_i$, иначе $Z_{ij} = 0$. Он представляет собой итеративную сеть g ячеек [8].

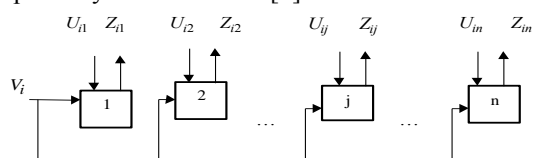


Рисунок 6 - Операционное устройство модуля сравнения

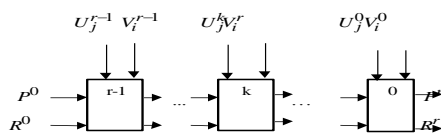


Рисунок 7 – Блок j модуля сравнения

На рисунке 8 обозначено: U_{ij}^k – разряд k числа U_{ij} ; V_i^k – разряд k числа V_i ; P^r, R^{r-1} – входные граничные сигналы: $P^r = 1, R^r = 0$; P^0, R^0 – выходные граничные сигналы.

Входные граничные сигналы вырабатывает локальное устройство управления. Ячейка k выполняет логические функции сравнения чисел U_{ij} и V_i : $P^{k-1} = P^k \& (U_{ij}^k \vee \bar{V}_i^k)$; $R^{k-1} = R^k \vee P^k \& U_{ij}^k \& \bar{V}_i^k$; если $P^0 \vee R^0 = 1$, то блок формирует признак $Z_{ij} = 1$, иначе $Z_{ij} = 0$.

Модуль умножения формирует элементы вектора P_i : $P_{ij} = F_{ij} \cdot V_i$. Модуль имеет локальную входную и выходную память, операционное устройство и устройство управления. Входная память модуля умножения хранит числа F_{ij} и V_i , $j=1,2,\dots,n$. Выходная память хранит произведения $P_{ij} = F_{ij} \cdot V_i$.

Операционное устройство, показанное на рисунке 8, состоит из n блоков, работающих параллельно. Блок j является матричным одноктактным умножителем [8]. Он выполняет умножение чисел F_{ij} и V_i по методу умножения, начиная со старших разрядов множителя, со сдвигом множимого вправо.

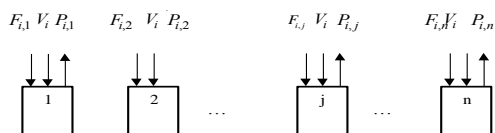


Рисунок 8 – Операционное устройство модуля умножения

Модуль деления формирует элементы вектора L_i : $L_{ij} = P_{ij} / B_j$. Модуль имеет локальную входную и выходную память, операционное устройство и устройство управления. Входная память модуля деления хранит числа P_{ij} и B_j , $j=1,2,\dots,n$. Выходная память хранит числа $L_{ij} = P_{ij} / B_j = F_{ij} \cdot V_i / B_j$.

Операционное устройство модуля деления, показанное на рисунке 9, состоит из n блоков, работающих параллельно. Блок j является матричным одноктактным устройством деления на основе итеративной сети [9]. Он выполняет целочисленное деление числа P_{ij} на число B_j по методу деления с незавершением (nonperforming division).

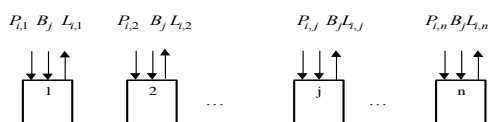


Рисунок 9 – Операционное устройство модуля деления

Счетный модуль используется для подсчета количества размещенных копий файла и формирования сигнала W , управляющего формированием признаков Y_{ij} . Входная память модуля хранит значение C_i . Структура операционного устройства счетного модуля показана на рисунке 10. Она имеет элемент ИЛИ, счетчик СТ, схему сравнения, триггер.

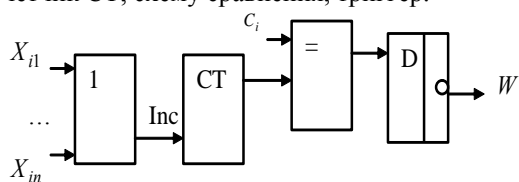


Рисунок 10 - Операционное устройство счетного модуля

Схема формирования признаков показана на рисунке 11. Она вырабатывает сигналы Y_{ij} : $Y_{ij} = -(W \& X_{ij}) \& Z_{ij} \& RD$, $j=1,2,\dots,n$. Значение $Y_{ij} = 0$ есть признак того, что узел j запрещен для размещения файла i .

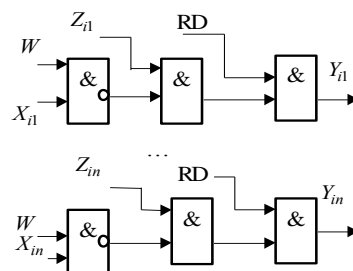


Рисунок 11 – Схема формирования признаков

Модуль поиска максимума (МПМ) используется для определения узлов с наибольшим значением L_{ij} . Модуль имеет локальную входную и выходную память, операционное устройство и устройство управления. Входная память модуля хранит числа L_{ij} . Операционное устройство модуля поиска максимума является итеративной сетью $n \times r$ ячеек [8], показанной на рисунке 12. Ячейка на пересечении строки j и столбца k выполняет логические функции сравнения L_{ij} : $P_j^{k-1} = P_j^k \& (L_{i,j}^k \vee R^k_j)$; $Q^{k}_{j+1} = Q^k_j \vee L_{i,j}^k \& P_j^k$; $R^{k}_{j+1} = R^k_j$; $R^k_1 = \bar{Q}^k_{n+1}$, $j=1,2,\dots,n$; $k=0,1,\dots,r-1$. Входные граничные сигналы вырабатывает локальное устройство управления: $P_j^r = 1$; $Q_1^k = 0$. Если $P_j^0 = 1$, то $X_{ij} = 1$, иначе $X_{ij} = 0$.

Для выделения первого среди наибольших значений L_{ij} используется селектор ведущей единицы (СВЕ). Он является итеративной сетью из n ячеек [10], показанной на рисунке 13. Ячейка j выполняет функции: $Y_j = P_j^0$, $X_{ij} = P_j^0 \& \bar{Y}_{j+1}$.

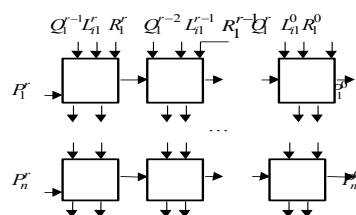


Рисунок 12 – Операционное устройство модуля поиска максимума

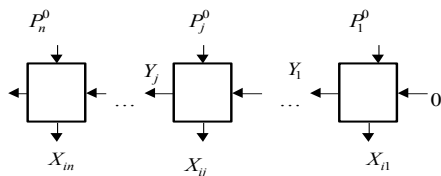


Рисунок 13 – Структура селектора ведущей единицы

Счетный модуль, схема формирования признаков, модуль поиска максимума и селектор ведущей единицы предназначены для выполнения процедуры формирования строки X_i : $X_{ij} = 1$, если файл i должен размещаться в узле j , иначе $X_{ij} = 0$.

Процесс формирования элементов вектора X_i происходит следующим образом. На первом шаге значения X_{ij} равны нулю, $Y_{ij} = 1$ и счетчик СТ установлен в нуль. На втором шаге содержимое счетчика сравнивается с C_i . Если они не равны между собой, вырабатывается сигнал W . С его помощью на третьем шаге формируются новые значения Y_{ij} :

$Y_{ij} = \neg(W \& X_{ij}) \& Z_{ij} \& RD$. На четвертом шаге осуществляется поиск первого максимального значения среди значений L_{ij} , таких, что $Y_{ij} = 1$.

В результате этого, одно из значений X_{ij} устанавливается в единицу. На пятом шаге содержимое счетчика увеличивается на единицу, и процесс повторяется с шага 2 до тех пор, пока $W = 1$. Для того чтобы определенные на данном шаге значения X_{ij} не потерялись на следующем шаге, при их сохранении используется блок элементов ИЛИ. Если все значения X_{ij} сформированы, то сигнал W устанавливается в ноль. Структура для формирования строки X_i показана на рисунке 14.

Модуль вычитания формирует элементы вектора U_{i+1} : $U_{i+1,j} = U_{ij} - V_i X_{ij}$. Модуль имеет локальную входную и выходную память, операционное устройство и устройство управления. Операционное устройство модуля вычитания показано на рисунке 15. Оно состоит из n блоков, работающих параллельно. Входная память модуля состоит из n – разрядного регистра значений X_{ij} , а также r – разрядных регистров, хранящих числа U_{ij} и V_i , $j=1,2,\dots,n$.

Выходная память состоит из r – разрядных регистров, хранящих значения $U_{i+1,j}$.

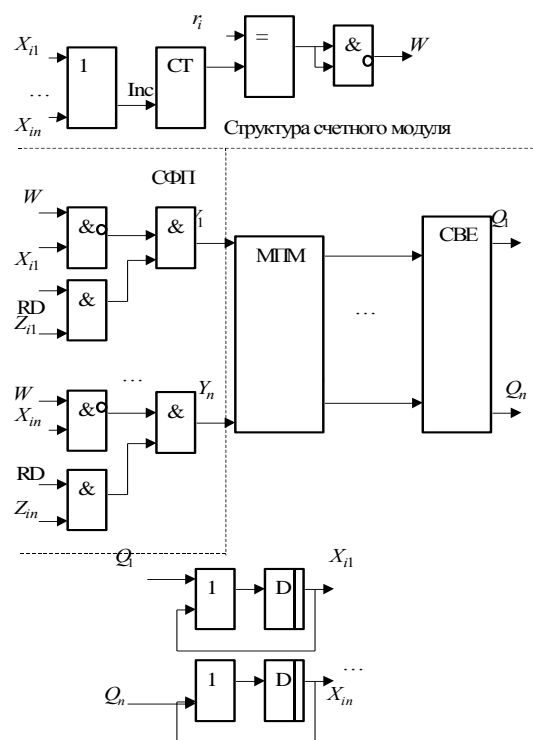


Рисунок 14 - Структура для формирования строки X_i

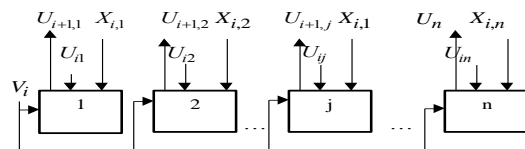


Рисунок 15 - Операционное устройство модуля вычитания

Блок j , показанный на рисунке 17, содержит схему поразрядной конъюнкции (СРК) и вычитатель r -разрядных чисел. СРК выполняет логическую функцию $V_i^k \& X_{i,j}$, где V_i^k - разряд k числа V_i . Блок работает следующим образом. Если $X_{ij} = 1$, то к вычитателю подается значение V_i . На выходе вычитателя формируется значение $U_{i+1,j}$: $U_{i+1,j} = U_{ij} - V_i X_{ij}$. Если $X_{ij} = 0$, то вместо значения V_i к вычитателю подается ноль и значение U_{ij} не изменяется.

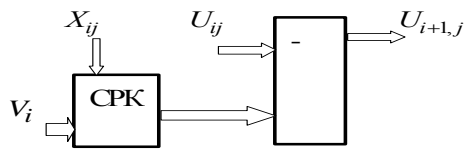


Рисунок 17 - Блок j модуля вычитания

Оценка эффективности спецпроцессора

Величина коэффициента эффективности спецпроцессора зависит от конкретной задачи размещения файлов. Обозначим: t_{cp} - время необходимое на одну операцию сравнения. Для размещения одного файла на однопроцессорном компьютере требуется в лучшем случае $n+1$ операций сравнения. Время необходимое на все операции сравнения в лучшем случае:

$T_1^n = (n+1)t_{cp}$. В худшем случае для размещения одного файла требуется $2n-1$ операций сравнения. Время необходимое на все операции сравнения в худшем случае: $T_1^x = (2n-1)t_{cp}$. Для размещения одного файла с помощью разработанной структуры спецпроцессора требуется 2 операции сравнения в любом случае:

$T_2 = 2t_{cp}$. Минимальное значение коэффициента ускорения от использования спецпроцессора:

$$S_{\min} = T_1^n / T_2 = (n+1)t_{cp} / 2t_{cp} = (n+1) / 2.$$

Максимальное значение коэффициента ускорения от использования спецпроцессора:

$$S_{\max} = T_1^x / T_2 = (2n-1)t_{cp} / 2t_{cp} = n-1/2.$$

Ускорение решения задачи размещения файлов при использовании спецпроцессора пропорционально количеству узлов компьютерной сети.

Выводы

Статья посвящена повышению эффективности работы компьютерной сети за счет оптимизации размещения файлов с применением специализированных структур. Для ускорения размещения файлов по узлам компьютерной сети предложена структура

спецпроцессора, ориентированная на решение этой задачи.

Литература

1. Бельков Д.В. Методы и вычислительные структуры для оптимизации размещения файлов в компьютерных сетях. Автореферат диссертации. Донецк: 2004. – 20 с.
2. F. Braun, J. Lockwood, M. Waldvogel. Protocol wrappers for layered network packet processing in reconfigurable hardware. //IEEE Micro - 2002. - № 1. - P. 12 – 19.
3. Braun F., Lockwood J., Waldvogel M. Layered protocol wrappers for Internet packet processing in reconfigurable hardware. //www.arl.wustl.edu/arl/projects/fpx/
4. Lockwood J., Naufel N., Turner J. S., Taylor D. E. Programmable network packet processing on the Field Programmable Port Extender (FPX). //www.arl.wustl.edu/arl/projects/fpx/
5. Ладыженский Ю.В., Бельков Д.В. Структура спецпроцессора для оптимизации размещения файлов по узлам вычислительной сети. // 36. Наукових праць ДонНТУ. Серія “Проблеми моделювання та автоматизації проектування динамічних систем”. Вип. 52: - Донецьк: ДонНТУ, 2002. - С. 119-124.
6. Декларацийний патент на винахід № 59736А, G 06 F 9/46 /Бельков Д.В., Ладыженський Ю.В., Пристрій для розподілу файлів серед комп'ютерів. Заявл. 29.11.2002. Опубл. 15.09.2003, Бюл. № 9.
7. Огнев И.В., Борисов В.В. Ассоциативные среды. М.: Радио и связь, 2000.– 309 с.
8. Фет Я.И. Параллельные процессоры для управляющих систем. М.: Радио и связь, 1981.–157 с.
9. Карцев М.А., Брик В.А. Вычислительные системы и синхронная арифметика. М.: Радио и связь, 1981.- 358 с.
10. Бандман О.Л. Специализированные процессоры для высокопроизводительной обработки данных. Новосибирск: Сибирское отделение АН СССР, 1988. – 289 с.

Бельков Д.В. Вычислительные структуры для размещения файлов по узлам компьютерной сети. Статья предлагает метод и вычислительные структуры для оптимизации размещения файлов в компьютерных сетях. Для повышения эффективности компьютерных сетей за счет рационального размещения файлов сформулирована задача рационального размещения файлов. Для ускорения решения задачи предложены вычислительные структуры, ориентированные на ее решение.

Ключевые слова: Файлы, узлы компьютерной сети, специализированные процессоры для размещения файлов.

Belkov D.V. Methods and computer structures for optimization of file allocation in the computer networks. The article presents the method and computer structures for optimization of file allocations in computer networks. For a rise of efficiency of computer network due to the rational file allocations, the task of the rational file allocations on nodes of computer network is formulated. Heuristic method for solving of this task is proposed. For acceleration of solving of task of file allocations on nodes of computer network, structures of special processors oriented on the solving of this task are proposed.

Keywords: Files, nodes of computer network, special processors for file allocation.

Статья поступила в редакцию 20.09.2017

Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

УДК 004.0

Модель нечеткой логики в функциональном алгоритме управления информацией и коммуникациями предприятий

Боднар А.В., Николаенко Д.В.

ГОУВПО «Донецкий национальный технический университет»

Боднар А.В., Николаенко Д.В., Модель нечеткой логики в функциональном алгоритме управления информацией и коммуникациями предприятий. Статья посвящена актуальным вопросам управления развитием коммуникаций предприятия. Проанализированы основные принципы управления. Определена взаимосвязь функций и методов управления коммуникациями, а так же последовательность трансформаций функций, методов и средств в управленческие решения. Разработан механизм управления информацией и коммуникациями. Представлен усовершенствованный функциональный алгоритм управления информацией и коммуникациями на основе нечеткого моделирования.

Ключевые слова: алгоритм, коммуникации, модель, управление, нечеткая логика

На сегодняшний день информация и коммуникации приобретает все большее значение для сохранения стабильности работы предприятий в связи с постоянно изменяющейся внешней средой, которая требует незамедлительной ответной реакции предприятия в виде соответствующих управленческих решений. Однако, управление коммуникациями носит бессистемный характер что существенно снижает эффективность работы предприятий, не уделяется должное внимание актуализации информации о внутренней и внешней среде в режиме реального времени, коммуникации воспринимаются как вспомогательные функции, что влечет за собой конфликты между уровнями управления, затрудняет их интеграцию в систему управления предприятием, а так же не позволяет принимать оперативные управленческие решения. Недооценка важности организационных коммуникаций и отсутствие методологических подходов к формированию функционального алгоритма управления информацией и коммуникациями, приводят к нерациональному использованию ресурсов, затрачиваемых на обеспечение обмена информацией внутри предприятия и с внешней средой, и как следствие, к неэффективному управлению предприятием в целом.

Вопросами управления информацией и коммуникациями занимались такие отечественные и зарубежные авторы, как Н. Морозова [1], Ю. Лысенко [2], А. Леоненков [3], С. Патрицки [4], С. Ильяшенко [6] которые освещали проблемы снижения эффективности процесса коммуникации предприятий, систематизировали знания о системе управления коммуникациями. Однако на сегодняшний день

недостаточно сформировано понимание важности применения алгоритмов управления системой коммуникаций на отечественных предприятиях, не существует общей методологии и концептуально сформулированного подхода их формирования и реализации. В связи с этим открытыми остаются вопросы анализа возможных способов принятия своевременных управленческих решений, критериев и методики оценки их эффективности, а так же организационных подходов к управлению коммуникациями.

Цель исследования заключается в формировании функционального алгоритма управления информацией и коммуникациями на основе модели нечеткой логики.

Важным направлением совершенствования системы управления предприятия становится решение проблем разработки теоретико-практических основ и методологических подходов к применению механизма управления информацией и коммуникациями, основанного на современных экономико-математических методах, а так же его внедрение в контур управления предприятием в условиях кризиса.

С этой целью необходимо усовершенствовать механизм управления коммуникациями, который включает в себя принципы, методы, средства, функции и управленческие решения, направленные на достижение поставленных целей.

Основными принципами управления коммуникациями являются:

- принцип формирования структуры коммуникаций, отвечающей потребностям сотрудничества, партнерства и компетентности;

- принцип регулирования масштабов коммуникаций для определенных должностных лиц и подразделений;

- принцип информационного обеспечения коммуникаций, распределение информации для установления и поддержания необходимых коммуникаций;

- принцип регулирования коммуникативной нагрузки различных звеньев организации;

- принцип учета неформальных коммуникаций и использования их в процессах управления;

- принцип локализации негативных коммуникаций и снижения их влияния на деятельность людей;

- принцип полномочий и ответственности;

- принцип рационального использования рабочего времени руководителя;

- принцип целостности;

- принцип однозначности;

- принцип иерархичности [1].

Данные принципы пронизывают все стадии управления, являясь основой формирования механизма управления коммуникациями.

В свою очередь механизм управления коммуникациями базируется на реализации пяти основных функций управления: планирование, организация, мотивация, контроль и регулирование. Набор этих функций пронизывает все его этапы, а так же неразрывно связан с методами управления коммуникациями, которые включают в себя совокупность возможных способов влияния управляющей системы на управляемую, необходимых для достижения поставленных целей.

Методы управления коммуникациями – это способы реализации функций менеджмента, которые в соответствии с традиционной классификацией могут быть разделены на организационные, экономические, социально-психологические, административные. Однако, такое группирование является достаточно условным, так как они имеют много общих черт и могут взаимодополнять друг друга [2].

Методы управления коммуникациями неразрывно связаны со средствами управления, которые являются инструментами или рычагами воздействия управляющего объекта на управляемый. Примеры средств управления, которые могут быть использованы при управлении коммуникациями, в рамках соответствующих функций управления.

Процесс управления – это целенаправленное воздействие субъекта управления на объект управления посредством последовательной реализации функций управления, направленных на достижение поставленных целей предприятия и может в целом рассматриваться как процесс принятия управленческих решений. То есть, имеет место взаимосвязь функций управления и процедур принятия решений, которые являются трудноотделимыми друг от друга видами управленческой деятельности [3].

Элементами механизма управления коммуникациями являются объект управления, а так же цели, функции, методы и структуры субъекта управления.

Поскольку механизм управления коммуникациями формируется на основании системного подхода, определим его объект и субъект. Объектом управления является система коммуникаций, включающая социальную, управленческую, информационную, техническую и организационную составляющие.

Субъектом управления выступают руководители управления коммуникациями, каждый из которых воздействует на них путем реализации своих функций. Субъект управления вырабатывает управляющее воздействие в виде приказа, команды, сигнала, которое передается объекту управления

В связи с этим, результатом трансформации методов, функций и средств управления является управленческое решение, выработанное в соответствии с основными принципами управления, которое может служить заключительным этапом работы механизма управления коммуникациями.

Управленческое решение представляет собой выбор альтернативы из всех возможных, направленной на достижение целей предприятия с наименьшими затратами.

Процесс принятия решения состоит из последовательности этапов и процедур, которые можно представить следующим образом:

- диагностика и формулирование проблемы;

- формулирование ограничений и формулирование критериев принятия решений;

-определение альтернатив;

- оценка альтернатив;

- выбор альтернатив;

- реализация решения;

- принятие решения (обратная связь).

Однако степень неопределенности в проблеме управления системой коммуникаций существенна, так как многие оценки носят нечеткий характер. Именно по этой причине была предложена идея усовершенствования

механизма управления коммуникациями, на основе применения нечеткой модели, которая позволит принимать оптимальные управленческие решения [4].

Механизм управления коммуникациями является составляющей частью внутренней среды предприятия, и его специфика соответствует особенностям конкретного субъекта управления. Каждый элемент механизма по отдельности не обладает теми свойствами, которыми обладает их совокупность, т.е. данному механизму присуща эмерджентность. Иначе говоря, механизм управления, как совокупность определенных элементов, обладает другим набором свойств, не присущим его подсистемам и блокам.

Главная задача механизма управления коммуникациями – организация процесса управления коммуникациями предприятия. В соответствии с главной задачей, назначение механизма управления коммуникациями должно заключаться в следующем:

- формирование системы управления коммуникациями предприятия и обеспечение ее адекватности складывающимся внешним и внутренним условиям деятельности;
- принятие оперативных управленческих решений;
- обеспечение эффективной циркуляции информации в рамках коммуникационного процесса;
- реализацию стимулов, направленных на повышение эффективности коммуникаций;
- распределение обязанностей между субъектами управления коммуникациями.

Механизм управления коммуникациями представляет собой открытую систему со входом и выходом. На входе в механизм поступают ресурсы, на выходе происходит передача во внутреннюю и внешнюю среду неиспользованных и преобразованных ресурсов. Одним из вариантов выхода работы механизма является управленческое воздействие. В качестве объекта управления механизма выступает система коммуникаций (социальная, управленческая, информационная, техническая и организационная составляющие).

На рисунке 1 представлена нечеткая продукционная модель механизма управления коммуникациями. К основным компонентам продукционной модели добавлена система коммуникаций предприятия, предоставляющая исходные данные для блока фаззификации. Управленческое решение, сформированное на выходе блока дефаззификации позволяет оказать управленческое воздействие на систему коммуникаций в целом [5].

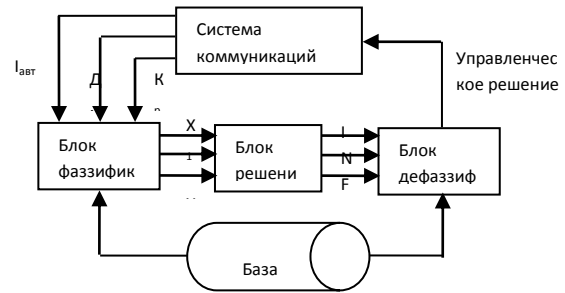


Рис. 1. Механизм управления информацией и коммуникациями на основе нечеткого моделирования

В отдельный блок вынесена база знаний, лингвистические переменные и нечеткие правила, которые позволяют формировать нечеткий вывод и приводить полученные результаты к четкости, т.к. к некоторым числовым значениям, находящимся в заданных рамках изменения.

Основой работы организационно-экономического механизма управления информацией и коммуникациями является функциональный алгоритм, который отображает последовательность действий, начиная с выбора и расчета входящих коэффициентов и до получения управленческого решения и разработки средств по усовершенствованию системы коммуникации. Реализацию механизма, основанного на процессе композиционного нечеткого вывода, предлагается производить при помощи инструментального средства MATLAB пакетом Fuzzy Logic Toolbox.

Началом работы алгоритма является постановка задачи, которая может быть сформулирована следующим образом: необходимо оценить уровень развития системы коммуникаций предприятия для выявления достаточности ее возможностей по отношению к задачам управления (рис. 2).

С этой целью происходит формирование системы показателей, которые характеризуют состояние ситуации в зависимости от факторов определяющих ее развитие. Для этого в работе было выделено 15 входящих показателей, характеризующих пять составляющих системы коммуникаций:

- информационная (автоматизация получения входящей информации, автоматизация документооборота, потери информации);
- техническая (достаточность программного обеспечения, достаточность аппаратного обеспечения, загрузка каналов связи);

- управленческая (централизация коммуникаций, детализация информации, динамичность коммуникаций);
- организационная (обратная связь, реализация долгосрочных целей, реализация текущих заданий);
- социальная (необходимость повышения квалификации, текучесть кадров, уровень конфликтности).

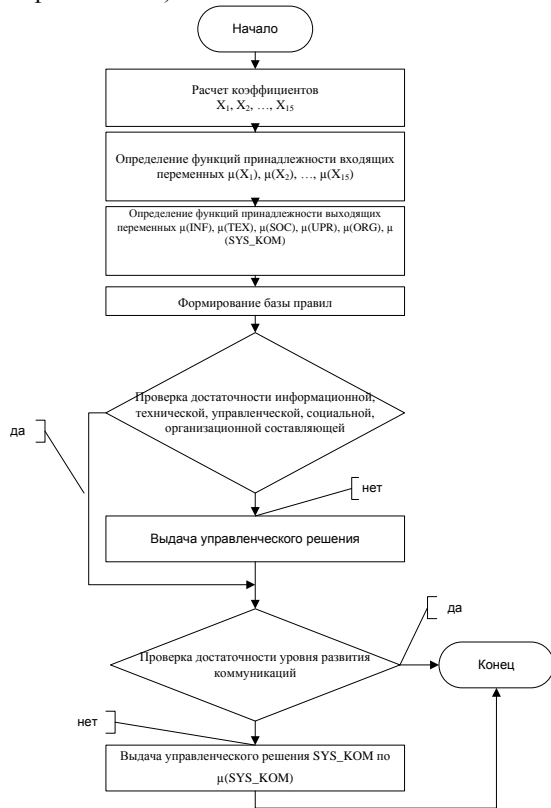


Рис. 2. Функциональный алгоритм организационно-экономического механизма управления информацией и коммуникациями, основанный на применении теории нечеткой логики

После формирования всех входящих и выходящих переменных модель имеет следующий вид в редакторе систем нечеткого вывода FIS (рис. 3).

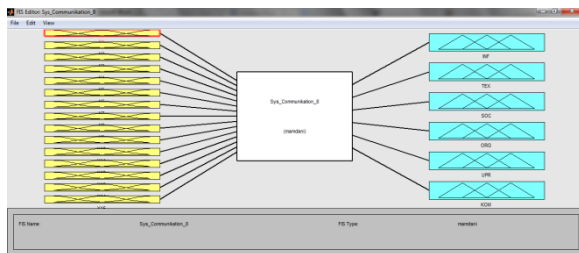


Рис. 3. Входящие и выходящие показатели в редакторе систем нечеткого вывода FIS

Реализованная нечеткая модель функционального алгоритма управления системой коммуникаций позволяет подставить конкретные значения для каждого из определенных параметров по исследуемому предприятию.

На основании этих входных данных программа генерирует нечеткий результат обобщения, а так же рассчитывает четкое значение искомого показателя методом дефаззификации.

На рис. 4 изображен вывод нечеткого результата по каждой из выходных переменных при оценке входных параметров ПАО «Альфа-банк».

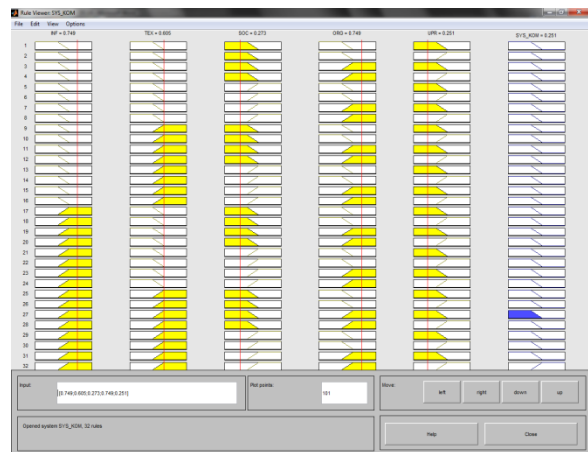


Рис. 4. Результат анализа уровня развития системы коммуникаций ПАО «Альфа-банк»

Анализируя представленные результаты ПАО «Альфа-банк» необходимо отметить, что система коммуникаций не находится на достаточном уровне развития, о чем свидетельствует низкое значение выходящего показателя $SYS_KOM = 0.251$.

Это связано с тем, что техническая составляющая требует принятия управленческих решений для повышения эффективности работы системы коммуникации предприятия в целом.

На рис. 5. представлена зависимость выходящей переменной от технической и управленческой составляющих ПАО «Альфа-банк»

Представленная зависимость отражает равное влияние выбранных составляющих, технической и управленческой на результирующий показатель уровня развития системы коммуникаций.

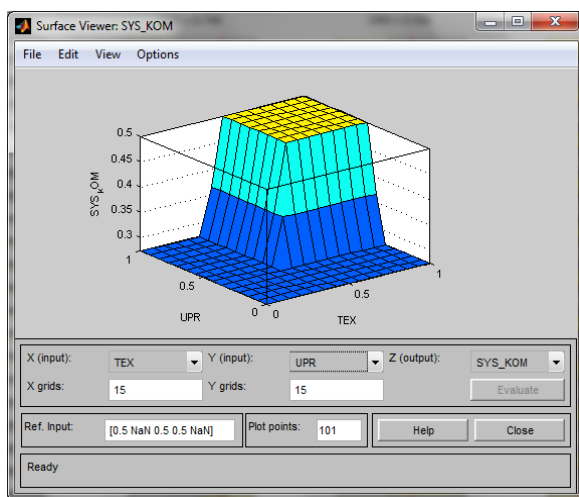


Рис. 5. Зависимость выходящей переменной от технической и управленческой составляющих ПАО «Альфа-банк»

В связи с тем, что техническая составляющая системы коммуникаций ПАО «Альфа-банк» продемонстрировала недостаточный уровень развития системы коммуникаций, было предложено управленческое в виде внедрения новой информационной системы, позволяющей обрабатывать запросы клиентов в одной программе.

Заключение

Таким образом, в статье рассмотрены теоретико-методологические аспекты развития информации и коммуникаций на предприятии. Представлена трансформация функций, методов и средств в управленческие решения, которая базируется на взаимосвязи функций и методов управления. Представлен усовершенствованный механизм управления информацией и коммуникациями предприятия и функциональный алгоритм на основе теории нечеткой логики. Рассмотрен методологический подход к формированию механизма управления коммуникациями.

Литература

1. Морозова Н.А. Управление коммуникациями в организации / Вестник ВГУ. Серия: Экономика и управление. – М. : 2010 – №2 – С. 173-180
2. Нечеткие модели и нейронные сети в анализе и управлении экономическими объектами: монография / под ред. чл.-кор. НАН Украины, д-ра экон. наук, проф. Ю.Г. Лысенко; Донец. нац. ун-т. – Донецк: Юго-Восток, 2012. – 386 с.
3. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. – Спб.: БХВ-Петербург, 2005. – 736 с.
4. Патрицки С. А. Пляскина Информационно коммуникационные технологии как инструмент трансформационных отношений в глобальном мире / Экономические инновации. – 2011. – № 43. – С. 253–260.
5. Инновации в маркетинге и менеджменте: монография / под ред. д.е.н., проф., С.М. Ильяшенко. – Сумы : ТОВ «Издательский дом «Папирус», 2013. – 616 с.
6. Data Acquisition and Signal Processing for Smart Sensors / Nikolay V. Kirianaki, Sergey Y. Yurish, Nestor O. Shpak, Vadim P. Deynega. – John Wiley & Sons, LTD Baffins Lane, Chichester, West Sussex, P019 IUD, England, 2001. – 275 p.
7. Information economy report 2013 ICTs, economics of cloud technologies and developing countries / United nations conference on trade and development UNCTAD. – New York and Geneva, 2012. – 115 p.
8. Kupritz, Virginia W.; Cowell, Eva. «Productive Management Communication: Online and Face-to-Face» Journal of Business Communication, – 2011, Vol. 48 Issue 1. – p54-82.

Боднар А.В., Николаенко Д.В., Модель нечеткой логики в функциональном алгоритме управления информацией и коммуникациями предприятий. Статья посвящена актуальным вопросам управления развитием коммуникаций предприятия. Проанализированы основные принципы управления. Определена взаимосвязь функций и методов управления коммуникациями, а так же последовательность трансформаций функций, методов и средств в управленческие решения. Разработан механизм управления информацией и коммуникациями. Представлен усовершенствованный функциональный алгоритм управления информацией и коммуникациями на основе нечеткого моделирования.

Ключевые слова: алгоритм, коммуникации, модель, управление, нечеткая логика

Bodnar AV, Nikolaenko DV, Fuzzy Logic Model in the Functional Algorithm of Information Management and Enterprise Communications. The article is devoted to topical issues of managing the development of enterprise communications. The basic principles of management are analyzed. The interconnection of functions and methods of managing communications, as well as the sequence of transformations of functions, methods and means into managerial decisions is defined. The mechanism of information and communication management has been developed. An improved functional algorithm for information and communication management based on fuzzy modeling is presented.

Keywords: algorithm, communication, model, control, fuzzy logic

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 65.011.42

Применение критериев принятия решений и теории игр при управлении экономическими показателями металлургического предприятия

В.А. Гуляев, Ю.К. Орлов
Донецкий национальный технический университет
gvard12@gmail.com

Гуляев В.А., Орлов Ю.К. Применение критериев принятия решений и теории игр при управлении экономическими показателями металлургического предприятия. Рассмотрена проблема принятия решений, обусловлен выбор типа базисного предприятия. Сформулирована задача, показывающая применение критериев для принятия решений. Произведен расчёт коэффициентов по имеющимся финансовым показателям предприятия, необходимых для решения задачи. Составлено регрессионное уравнение для выражения требуемого параметра через управляемые. Составлены матрица игры и матрица рисков. Произведены расчеты критериев. Выдвинуты предложения по программной реализации решения подобных задач на производстве.

Ключевые слова: решение, критерий, ЛПР, финансовая отчётность, регрессионное уравнение, матрица

Введение

Проблема принятия оптимальных решений является одним из путей к рационализации любого производства. Согласно теории управления, на вход «черного ящика» могут поступать входы, которые могут быть как неуправляемы, так и поддаваться изменению. На практике такие входы не являются полностью независимыми и прямо либо косвенно одни влияют в той или иной мере на некоторые другие. Как показывает опыт довольно часто необходимые показатели не могут изменяться напрямую, поэтому перед лицом принимающим решение (ЛПР), стоит задача найти такие значения, оценить степень корреляции с требуемым выходным параметром и определить такой уровень воздействия на управляемые показатели, чтобы получить заданный результат. Разработанные общепринятые критерии принятия решений в совокупности позволяют дать ЛПР возможность перебрать доступные варианты оказания влияния на входные параметры путем изменения какого-то конкретного параметра.

Металлургическое предприятие было выбрано в качестве базиса для исследования поскольку процессы, которые в нём протекают, позволяют сделать вывод о том, что это сложная система с множествами входов и выходов,

следовательно, результаты управления могут быть менее предсказуемыми, чем, например, у супермаркета. Также такой выбор обуславливает актуальность этой работы, поскольку принятие эффективных решений в подобных крупных предприятиях позволяет увеличивать прибыль в более значимых масштабах.

Применение достижений теории игр для принятия решений

В качестве дальнейшего теоретического изучения вопроса применения теории игр для принятия решений следует обратить внимание на работу Дж. Ч. Харшани «Игры с неполной информацией, сыгранные байесовскими игроками». Она стала продолжением работ известного математика Дж. Ф. Нэша, лауреата нобелевской премии за работу «Анализ равновесия в теории некооперативных игр». В этой работе и в других статьях он исследовал проблему выбора оптимальной стратегии поведения с учётом заранее известных стратегий других игроков. В свою очередь Харшани рассмотрел случай, когда стратегии и решения других игроков неопределенны. Ученый исходил из того, что каждый игрок является одним из нескольких «типов», а каждый тип отвечает набору возможных преимуществ для игрока и вероятно распределяет почти всех на типы игроков. Значит, каждый игрок в игре с неполной

информацией выбирает стратегию одного из таких типов. С согласованным требованием в отношении возможности распределения игроков Дж. Ч. Харшани показал, что для каждой игры с неполной информацией существует эквивалентная игра с полной информацией. То есть он трансформировал игру с неполной информацией в игру с несовершенной информацией. В таком случае игра может регулироваться моделями, предложенными Нэшем.

Также в этой области представляют интересы работы Р. Дж. Р. Селтена. В статье «Модель олигополии с инерцией спроса» Р. Дж. Р. Селтен разработал «чистую стратегию» с интуитивным выбором». Последовательно усложняя и уточняя отмеченное «равновесие» дополнительными условиями для предыдущих договоренностей об игре, ученый развивал ее с точки зрения динамики и приближал к условиям реальной жизни. Он на противоположных примерах доказал, что даже точки равновесия могут вызвать иррациональное поведение. По мнению ученого, только специальный класс точек равновесия (он их назвал «истинными», или «совершенными точками равновесия») обеспечивает на самом деле рациональное поведение в бескоалиционной игре (т.е. игре, которая не допускает свободного обмена информацией и принудительных условий, Нэш ей противопоставил кооперативные игры как такие, допускающие свободный обмен информацией и принудительные условия между игроками).

Формулировка задачи принятия решений

Допустим, что ЛПР поручается определить на Енакиевском металлургическом заводе (ЕМЗ) такой уровень текущей ликвидности, при котором значение коэффициента рентабельности оптимально для возможных колебаний коэффициента ликвидности. Ему следует учесть, что в условиях политической и экономической нестабильности, отражающей современные условия, величина краткосрочных обязательств необходимая для расчёта коэффициента текущей ликвидности, подвержена различным рискам изменения в большую сторону [1, 2]. В его распоряжении есть данные, необходимые для расчёта этих двух коэффициентов за период с IV квартала 2013 по III квартал 2015 года.

Данная задача была сформулирована на основании показателей, которые используются международным рейтинговым агентством Moody's в [3] для оценки предприятий горно-металлургической отрасли, а предприятие и

периодизация доступных данных – исходя из возможности получения объективных показателей из финансовых отчетностей, находящихся в открытом доступе. Данное агентство принадлежит к так называемой «большой тройке», которая включает в себя такие организации, как Standard & Poog's, Moody's и Fitch, вместе они составляют 94 процента рынка рейтингово-кредитных услуг на момент 2008 года. С точки зрения данной работы использование коэффициентов, употребляемых такой известной компанией повышает актуальность работы. К тому же, компания Moody's составляла отчетность по состоянию исследуемого предприятия в 2011 году, что еще раз подчеркивает реальность подобной задачи на практике.

Подготовку к решению поставленной задачи было принято решение условно разбить на следующие подпункты:

- а) расчёт заданных коэффициентов на временном промежутке;
- б) получение регрессионного уравнения, описывающего степень зависимости управляемого параметра от требуемого;
- в) генерация массива решений;
- г) расчёт матриц игры и рисков [4].

После выполнения описанных пунктов можно приступать к непосредственному расчёту критериев принятия решений.

Расчёт коэффициентов

Прежде чем считать сами коэффициенты необходимо определить формулы для их расчёта, которые выглядят следующим образом

$$\text{рентабельность} = \frac{\text{ф.2стр.2355}}{\text{ф.1стр.1300}} \quad (1)$$

$$\text{ликвидность} = \frac{\text{ф.1стр.1195}}{\text{ф.1стр.1695}} \quad (2)$$

Формулы (1) и (2) представлены в виде, используемом в экономических науках, где, например, «ф.2 стр. 2355» обозначает вторую форму бухгалтерской отчетности, который называется «Отчёт о финансовых результатах», из которой берется показатель с индексом 2355, который в данном случае обозначает чистый финансовый убыток. В зависимости от страны ведения отчётности индексы могут меняться, но содержания отчета в целом стандартизировано. В данной работе используются отчетности, составленные по украинским стандартам. Далее на основании полученных финансовых отчетностей и формул (1) и (2), взятых из [5] можно выполнить расчёт коэффициентов, который показан в таблице 1.

Таблица 1. Расчёт используемых финансовых коэффициентов

| Период | Рент. | Ликв. |
|------------|---------|--------|
| 4 кв. 2013 | -0,0261 | 0,8615 |
| 1 кв. 2014 | -0,0430 | 0,8246 |
| 2 кв. 2014 | -0,0332 | 0,8388 |
| 3 кв. 2014 | -0,0567 | 0,8007 |

Продолжение таблицы 1

| Период | Рент. | Ликв. |
|------------|---------|--------|
| 4 кв. 2014 | -0,0181 | 1,0031 |
| 1 кв. 2015 | 0,0160 | 1,0551 |
| 2 кв. 2015 | -0,0002 | 1,0430 |
| 3 кв. 2015 | -0,0261 | 1,0097 |

Данные показатели свидетельствуют о том, что исследуемое предприятие имеет кризисные явления, поскольку, например, коэффициент рентабельности на всем промежутке отрицательный, что говорит об убыточности компании [6]. Подобная выборка для принятия наиболее оптимального решения не должна использоваться, поскольку она не учитывает общую динамику показателей, на которые могут повлиять прямо или косвенно такие параметры, которые неодинаково реагируют на изменения текущего состояния предприятия. В данном случае ввиду ограниченного доступа к финансовой информации могут быть использованы только эти параметры, что имеет и свои преимущества в виде потенциально более качественного решения в кризисных условиях.

Составление регрессионного уравнения

Далее необходимо составить регрессионное уравнение, описывающее зависимость между рассматриваемыми коэффициентами [7]. Высокий коэффициент корреляции между этими показателями даёт возможность говорить о том, что полученное регрессионное уравнение будет иметь не слишком большую погрешность. Для данного случая будет использоваться линейное регрессионное уравнение, принимающее вид

$$y = a + b \cdot x, \quad (3)$$

где y – коэффициент рентабельности;
 x – уровень текущей ликвидности;

a, b – коэффициенты регрессионного уравнения, которые нужно найти.

Для нахождения коэффициентов в формуле (3) необходимо было решить систему уравнений вида

$$\begin{cases} a \cdot n + b \cdot \sum x = \sum y \\ a \cdot \sum x + b \cdot \sum x^2 = \sum xy \end{cases}, \quad (4)$$

где n – количество рассматриваемых периодов, в данной задаче $n = 8$.

Подставив данные из таблицы 1 в формулу (4) получим систему вида:

$$\begin{cases} a \cdot 8 + b \cdot (-0,16) = 6,58 \\ a \cdot (-0,16) + b \cdot 0,00743 = -0,14 \end{cases}, \quad (5)$$

Данная система состоит из двух линейных уравнений и не представляет сложности в её решении. В результате вычислений получено, что $a = 1,0321$, $b = 4,0291$, подставив значения в формулу (3) получим искомое регрессионное уравнение

$$y = 1,0321 + 4,0291 \cdot x. \quad (6)$$

Таким образом, была получена формула для расчёта требуемого показателя через тот, который поддается управлению. Теперь при помощи него можно сгенерировать массив альтернативных решений и рассчитать относительно точное значение, которое получится при изменении управляемого параметра.

Генерация массива альтернатив. Составление матриц

Матричная игра – это парная игра, которая задается набором чистых стратегий $\{1, \dots, n\}$ и $\{1, \dots, m\}$ первого и второго игроков, а также платежной матрицей $(a_{ij})_{m \times n}$, определяющей выигрыш первого игрока при выборе игроками стратегий i и j соответственно. Целью первого игрока является максимизация своего выигрыша, а целью второго – минимизация выигрыша противника [8]. В контексте составленной задачи данное определение приобретает следующий вид: необходимо выбрать такую стратегию поведения, чтобы добиться наибольшего значения целевого параметра путём наименьшего воздействия на управляемый параметр.

Для выбора уровня оптимального уровня коэффициента текущей ликвидности, рассмотрены его значения на промежутке от -0,06 до 0,04 с шагом 0,02, т.о. получится 6 вариантов решений. Этот промежуток выбран на основании данных таблицы 3.2, где наименьшее значение показателя составило -0,0567 единиц, а наибольший – 0,0160 единиц, т.е. промежутки

немного выходит за рамки имеющихся данных. При помощи полученного уравнения (6) составлена матрица игры где каждый элемент сформирован по следующим правилам:

а) если фактическое значение коэффициента строго больше планового, то в матрицу заносится коэффициент рентабельности, рассчитанный по фактическому значению;

б) если фактическое значение коэффициента меньше планового, то в матрицу заносится коэффициент рентабельности, рассчитанный как среднее арифметическое планового и фактического значения.

Данные правила предложены в первую очередь для того, чтобы разнообразить элементы матрицы и на практике могут меняться.

В таблице 2 располагается матрица игры [8] по решаемой задаче.

Таблица 2 – Матрица игры

| | | Фактические значения | | | | | |
|-------------------|-------|----------------------|-------|-------|------|------|------|
| | | -0,06 | -0,04 | -0,02 | 0 | 0,02 | 0,04 |
| Плановые значения | -0,06 | 0,79 | 0,87 | 0,95 | 1,03 | 1,11 | 1,19 |
| | -0,04 | 0,83 | 0,87 | 0,95 | 1,03 | 1,11 | 1,19 |
| | -0,02 | 0,87 | 0,91 | 0,95 | 1,03 | 1,11 | 1,19 |
| | 0 | 0,91 | 0,95 | 0,99 | 1,03 | 1,11 | 1,19 |
| | 0,02 | 0,95 | 0,99 | 1,03 | 1,07 | 1,11 | 1,19 |
| | 0,04 | 0,99 | 1,03 | 1,07 | 1,11 | 1,15 | 1,19 |

Матрица рисков необходима для нахождения критерий Сэвиджа. На этапе планирования матрица составляется для того, чтобы получить общую картину по управлению ключевыми рисками процесса. Кроме того, в общем случае наличие матрицы позволяет оптимально распределять ресурсы и выстраивать правильную последовательность приоритетов. Её элементы рассчитываются по формуле

$$r_{ij} = \beta_j - a_{ij} \geq 0, \quad (7)$$

где r_{ij} – элемент матрицы рисков в i -строке j -столбца;

β_j – максимально возможное значение по факту;

a_{ij} – соответствующий элемент матрицы игры.

В соответствии с формулой (7) была сформирована матрица рисков [8] в таблице 3.

Как можно заметить, плановые и фактические значения альтернативы 0,04 при любом исходе являются безрисковыми, что может говорить уже сейчас о безальтернативности выбора этого параметра как наилучшего, однако на практике вряд ли получится такое простое решение, поэтому

необходимо продолжить процесс поиска оптимального решения.

Таблица 3 – Матрица рисков

| | | Фактические значения | | | | | |
|-------------------|-------|----------------------|-------|-------|------|------|------|
| | | -0,06 | -0,04 | -0,02 | 0 | 0,02 | 0,04 |
| Плановые значения | -0,06 | 0,20 | 0,16 | 0,12 | 0,08 | 0,04 | 0 |
| | -0,04 | 0,16 | 0,16 | 0,120 | 0,08 | 0,04 | 0 |
| | -0,02 | 0,12 | 0,12 | 0,12 | 0,08 | 0,04 | 0 |
| | 0 | 0,08 | 0,08 | 0,08 | 0,08 | 0,04 | 0 |
| | 0,02 | 0,04 | 0,040 | 0,04 | 0,04 | 0,04 | 0 |
| | 0,04 | 0 | 0 | 0 | 0 | 0 | 0 |

Расчёт критериев

Теперь можно приступить к непосредственному вычислению значений критериев и принятию решения. В данной работе будут рассчитаны следующие коэффициенты:

1. критерий Байеса;
2. критерий Лапласа;
3. критерий Вальда;
4. критерий Сэвиджа;
5. критерий Гурвица;
6. критерий Ходжа-Лемана.

Полный перечень формул для расчётов можно найти в [8, 9], поэтому их нецелесообразно приводить в данной работе ввиду большого объема. Используя эти формулы для расчёта коэффициентов, были получены следующие результаты и выбраны наилучшие решения, выделенные курсивом, что отражено в таблице 4.4.

Таблица 4 – Результаты расчётов критериев принятия оптимальных решений

| | Критерии | | | | | |
|-------|----------|-------|------|------|------|------|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| -0,06 | 0,95 | 0,99 | 0,79 | 0,20 | 0,95 | 0,88 |
| -0,04 | 0,96 | 0,99 | 0,83 | 0,16 | 0,97 | 0,90 |
| -0,02 | 0,97 | 1,01 | 0,87 | 0,12 | 0,9 | 0,93 |
| 0 | 1,00 | 1,03 | 0,91 | 0,08 | 1,02 | 0,96 |
| 0,02 | 1,03 | 1,05 | 0,95 | 0,04 | 1,04 | 1 |
| 0,04 | 1,07 | 1,093 | 0,99 | 0 | 1 | 1,04 |

Закончив все вычисления, ЛПР может принимать решения. Как видно на таблице 4, все критерии определяют вариант значения коэффициента текущей ликвидности равным 0,04 (поскольку ищется наибольшее значение по

альтернативам, кроме как для критерия Сэвиджа), что можно назвать ожидаемым результатом ввиду тривиальности задачи, т.е. знание личностных качеств ЛПР в данном случае является избыточной информацией. Полученные данные после оценки последствий принятого решения могут отправиться в базу данных для дальнейшего повторного использования при решении подобной задачи.

Возможные пути программной реализации решения

Рассматривая наиболее интересную с точки зрения внедрения программного продукта можно предложить концепцию программы, которая бы понимала такую задачу, как сформулированная в этой работе, т.е. понимала подтекст, которая могла бы брать необходимые данные из базы данных на серверах предприятия и самостоятельно давать оператору программы решение на естественном языке. Подобное программное решение могло бы иметь успех на рынке, однако ввиду сложности реализации можно допустить, что ЛПР должен выбрать тип решаемой задачи, задать начальные значения и принять самостоятельное решение на основании полученных расчётов.

В качестве путей реализации выделены следующие подходы:

- реализация на основе имеющихся на предприятии программных решений в виде надстроек, дополнений и прочего;
- реализация как отдельного продукта, который был бы написан на каком-либо языке программирования.

Данные подходы имеют преимущества и недостатки. У первого подхода существует определяющий недостаток – отсутствие опыта работы с системами предприятия у разработчика. Например, на ДМЗ внедрено решение от компании SAP, которое для написания дополнений к нему требует знания языка ABAP/4, который нигде больше кроме как в продуктах компании SAP не используется, а поскольку это ПО платное, то реализация этого подхода без доступа к системе будет проблематичной. Данный подход имеет один важный плюс – то, что система уже реализована на предприятии даёт возможность снизить расходы на переобучение персонала для работы с нововведениями, которые будут максимально эффективно работать на предприятии в дальнейшем. Также нужно упомянуть SAP Business Intelligence – подсистему, которая занимается обработкой имеющейся информации, в том числе нахождение требуемых показателей в режиме реального времени, линий трендов в

этих показателях с последующей визуализацией [10,11]. В целом в сравнении с SAP R/3 4.6, выпущенном в 2000 году и установленном на ДМЗ, данный программный продукт приобрёл значительное число функций, которые можно рассматривать как инструменты поддержки принятия решений, сюда входят в частности SAP IT Operations Analytics, SAP Analytics Hub, SAP BW4/HANA – все они позволяют, оперируя большими объемами данных, проводить аналитику текущего состояния предприятия.

Согласно отчёту IDC за 2015 год о результатах исследования российского рынка программного обеспечения информационных систем управления предприятием (ИСУП), или как их еще называют ERP-систем, более 80 процентов рынка занимают две компании – SAP и 1С (48,9 и 32,7 процента соответственно), тогда как Oracle занимала 4,4%, Microsoft 8,7%, Галактика – 2,1%. Также отмечается, что решение корпорации SAP преобладает в сегменте крупных предприятий, в то время как от 1С – в сегменте средних компаний. Металлургические заводы в целом относятся к крупным предприятиям (т.е. такие, у которых размер персонала превышает 500 человек), к тому же рассматриваемые предприятия уже внедрили решение SAP, поэтому рассмотрение других ERP-систем нецелесообразно.

Реализация в виде отдельного программного продукта, в свою очередь, компенсирует недостаток первого подхода, однако показывает другой – необходимость совместимости с имеющимся оборудованием, причем имеется ввиду не только требования к компьютерам по аппаратной части, но также и требования по приему и обработке данных, получаемых с датчиков (которые могут теоретически записываться в отдельную базу данных и использоваться в дальнейшем), что требует использования низкоуровневого программирования. Также, недостатками этого подхода является сложность по адаптации персонала к новому программному продукту и общая сложность проекта. В качестве ЯП, которые могут быть использованы при создании продукта, может быть выбран С# как современный инструмент, имеющий большое количество библиотек, в том числе и для управления различными датчиками, и позволяющий создать удобный интерфейс, тем самым снизив требования к переобучению персонала. В частности, для доступа к датчикам, установленным на ДМЗ, которые в большей мере используют Siemens Simatic для автоматизации сбора информации и управления технологическими процессами, можно

воспользоваться open-source библиотеками такими, как Libnodave или Snap7.

Важно также отметить преимущества от внедрения в такую программу нейросетей, которые могли бы обучаться на основании имеющихся данных и выдавать наилучшее решение с меньшей долей вероятности ошибки – этот подход может быть применен в целом и на предложенных подходах в той или иной мере.

Таким образом, выбор оптимального подхода остается за разработчиком и должен основываться на имеющихся возможностях информационной инфраструктуры предприятия и доступных денежных средствах.

Выводы

Таким образом, была исследована проблема принятия решений при управлении экономическими параметрами металлургического завода. Была решена задача, демонстрирующая применение теории игр для принятия решений. Исследованы текущие направления в развитии теории игр как научной дисциплины. Предложены варианты программной реализации решений таких задач, оценены преимущества и недостатки каждого из предложенных подходов.

Литература

1. Тагиев, К.Ф. Технологические инновации на машиностроительном предприятии: риски и пути их снижения / К.Ф. Тагиев // Вестник экономики, права и социологии, 2010. – №4. – С.64–69.
2. Ширинкина, Е.В. Прогнозирование риска несостоятельности предприятия в условиях неопределенности / Е.В. Ширинкина // Вестник Пермского университета, 2015. – № 3(26). – С.137–144.
3. Global Mining Industry. Rating methodology. – Moody's Investors Service, 2014. – 24 с.
4. Турсунов, Т.М. Рейтинговые агентства и их роль на

рынке структурированных финансовых инструментов / Т.М. Турсунов // Журнал «Корпоративные финансы», 2010. – №3(15) – С.79–83.

5. Литвак, Б.К. Стратегический менеджмент: учебник / Б.К. Литвак. – М. : Юрайт, 2015. – 507 с.
6. Шимко, А.К. Микроэкономика (в вопросах и ответах): учебно-методическое пособие / А.К. Шимко, Л.В. Бондарь, В.Н. Братухин, Т.Н. Гутовец и др. – Минск : Част. ин-т управ. и предпр, 2006. – 61 с.
7. Акимов, В.А. Основы анализа и управления рисков в природной и техногенной сферах / В.А. Акимов, В.В. Лесных, Н.Н Радаев. – М. : Деловой экспресс, 2004. – 352 с.
8. Блягоз, З.У. Принятие решений в условиях риска и неопределенности / З.У. Блягоз, А.Ю. Попова // Вестник Адыгейского государственного университета, 2006. – №4. – С.28–34.
9. Просветов, Г.И. Математические методы в экономике: учебно-методическое пособие / Г.И. Просветов. – М. : Изд-во РДП, 2004. – 364 с.
10. Трахтенгерц, Э.А. Компьютерные системы поддержки принятия управленческих решений / Э.А. Трахтенгерц // Проблемы управления, 2003. – №1. – С.13–28.
11. Гуляев В.А., Орлов Ю.К. Исследование влияния работы кислородно-компрессорного цеха на деятельность металлургического завода. – Информатика, управляющие системы, математическое и компьютерное моделирование в рамках III форума «Инновационные перспективы Донбасса» (ИУСМКМ - 2017): VIII Международная научно-техническая конференция, 25 мая 2017, г.Донецк: / - Донецк: ДонНТУ, 2017. – с. 718-725.

Гуляев В.А., Орлов Ю.К. Применение критериев принятия решений и теории игр при управлении экономическими показателями металлургического предприятия. Рассмотрена проблема принятия решений, обусловлен выбор типа базисного предприятия. Сформулирована задача, показывающая применение критериев для принятия решений. Произведен расчёт коэффициентов по имеющимся финансовым показателям предприятия, необходимых для решения задачи. Составлено регрессионное уравнение для выражения требуемого параметра через управляемые. Составлены матрица игры и матрица рисков. Произведены расчеты критериев. Выдвинуты предложения по программной реализации решения подобных задач на производстве.

Ключевые слова: решение, критерий, ЛПР, финансовая отчётность, регрессионное уравнение, матрица

Gulyaev V.A., Orlov J. C. Application of decision criteria and game theory in economic index management of a metallurgical facility. The problem of decision-making was considered, the choice of the type of basic facility was determined. A problem that shows the application of criteria for decision-making was formulated. The calculation of coefficients for the company's financial indexes required to solve the problem was made. A regression equation was constructed to express the required parameter via controllable one. The game matrix and the risk matrix were made. Calculations of criteria were made. Proposals on the program implementation of the solution of similar problems in the production were put forward

Keywords: solution, criterion, decision maker, financial statements, regression equation, matrix

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

Архитектура и сценарии распределенной обработки медицинских изображений веб-ориентированной PACS

Двойкин А. А., Привалов М. В., Воронова А. И.

Донецкий национальный технический университет, г. Донецк
кафедра автоматизированных систем управления
94dvoikin@gmail.com

Цель работы: повысить быстродействие веб-ориентированной системы архивирования, хранения, передачи и обработки медицинских изображений за счёт разработки архитектуры системы и сценариев распределённой обработки данных.

Введение.

При установлении диагноза и проведении лечения врачи всё больше полагаются на медицинские изображения, к которым относятся рентгенограмма, УЗИ, магнитно-резонансная томография, компьютерная томография, томография на позитивном излучении и т.д. Использование медицинских изображений непрерывно растёт по мере того, как во многих больницах (сначала в базовых больницах медицинских институтов) устанавливаются системы с высокой пропускной способностью для архивирования изображений. Медицинские изображения дают основной объём информации о пациенте, однако, самих по себе их недостаточно, поскольку требуется анализ и интерпретация в контексте истории болезни пациента (т.е. нужны метаданные, связанные с изображениями). Существует ряд факторов, которые значительно усложняют лечение пациента, основанное на медицинских изображениях: медицинские данные о пациентах собираются в разных лечебных учреждениях; врачи не всегда имеют доступ ко всем историям болезней всех своих пациентов; медицинские изображения представляются очень большими объёмами данных со сложной структурой. Объёмы обусловлены наличием 3D-изображений, показаний, полученных в последовательные моменты времени, многочисленных протоколов описания изображений, а сложность структур обусловлена клинически и эпидемиологически значимыми показателями, такими как возраст, питание, образ жизни и история болезни пациента, параметры получения изображений и анатомические и физиологические изменения. Часто изображений одной модальности оказывается недостаточно, поскольку на получение изображения влияет много

параметров, и дополнительная информация накапливается различными системами сбора физических данных (physical acquisition systems) [1].

Следовательно, информационные системы, которые предоставят возможность доступа к изображению с любого устройства и любого места при наличии Интернет-соединения, являются и будут оставаться актуальными ещё долгое время, ведь компьютерная обработка медицинских изображений постоянно развивается, появляются новые физические методы трёхмерного сканирования и визуализации внутренних органов человека [2].

Обзор существующих систем.

В данный момент существует достаточное количество систем архивации и передачи медицинских изображений (PACS – Picture Archiving and Communication Systems). Они решают следующую проблему: более широкое применение цифровой патологии в лаборатории означает, что размер изображений и метаданных, подлежащих хранению, продолжает увеличиваться, что приводит к увеличению спроса на места хранения. Наряду с увеличением требований к пространству для хранения данных скорость, с которой эти изображения необходимо анализировать и извлекать, также должна соответствовать растущим требованиям этого динамического диагностического поля [3]. Система должна быть масштабируемой и поддаваться регулярным обновлениям, учитывая быстрые изменения, которые разработчики делают с существующими решениями для анализа и хранения [4].

Для решения данной проблемы возможно использование сервис-ориентированных и распределённых Интернет-архитектур [5-9]. Однако, сервис-ориентированная архитектура на основе паттерна REST предлагает такие возможности: веб-ориентированность системы, не требующую больших затрат времени и громоздких задач по установке и настройке, простое и удобное добавление и модификацию новых алгоритмов разработчиками, доступ в

любое время и из любого места за счет веб-доступа [10].

Преимуществами таких систем является: наличие функционального веб-ориентированного клиентского приложения, возможность изменения и добавления новых алгоритмов в серверное приложение, возможность интеграции в системы анализа и управления изображениями и т.д.

Но основными и главными для данного исследования недостатками являются: отсутствие возможности распределения нагрузки между клиентскими терминальными устройствами и центром обработки данных (ЦОД) медицинского учреждения, а также добавления сервисов и расширения клиентского приложения. Так, например, открытие клиентского приложения на мобильном устройстве, которое не сможет произвести фрагментацию огромного изображения целиком, будет проблематичным, но, тем не менее,

обработка и отображение на экране какого-то фрагмента вполне возможны. Или если мы используем достаточно мощный для обработки ноутбук, но при этом кластеры ЦОД медицинского учреждения заняты, возможна ситуация, когда пользователь либо не получит изображение, либо будет ждать своей очереди.

Решением данной проблемы является разработка архитектуры, которая будет обладать высокой гибкостью и масштабируемостью, а также будет способна определять сценарий распределённой обработки изображений в зависимости от текущей ситуации.

Разработка архитектуры подсистемы хранения и обработки медицинских изображений

На рис. 1 изображена общая архитектура подобной подсистемы.

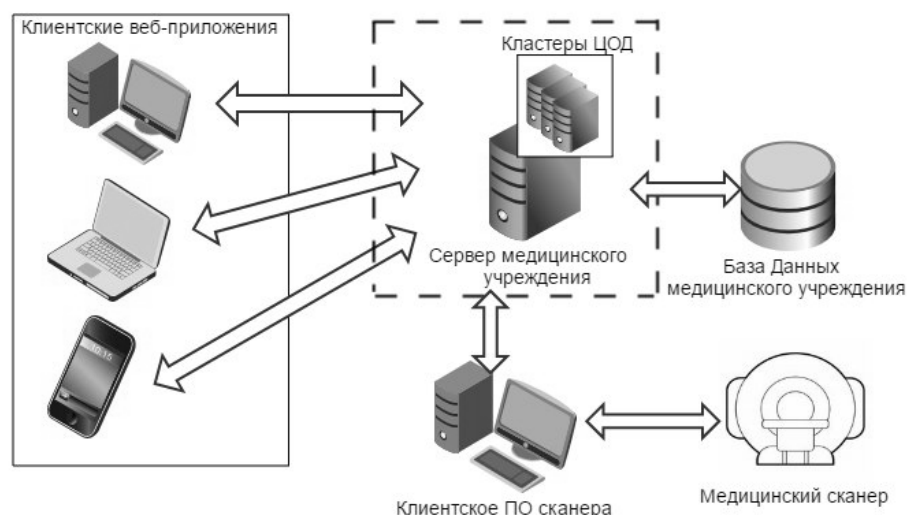


Рисунок 1 – Общая архитектура подсистемы хранения и обработки медицинских изображений

Из рис. 1 видно, что подсистема предоставит доступ к данным, которые хранятся в БД медицинского учреждения, с клиентских устройств благодаря взаимодействию клиентского веб-приложения с подсистемой, являющейся частью сервера медицинского учреждения. Данный сервер имеет прямой доступ к БД медицинского учреждения или удалённой БД, расположенной на кластере ЦОД. Более подробно архитектура подсистемы показана на рис. 2.

Взаимодействие между подсистемами основано на принципах сервис-ориентирования. Такие связи присутствуют между сервисами серверных подсистем и клиентским

приложением, аналогично построено взаимодействие и с DICOM-сервером.

Все сервисы подсистемы можно разделить на 5 основных групп:

- сервисы подготовки данных к передаче клиентскому приложению;
- сервисы обработки запросов клиентского приложения;
- сервисы динамического определения сценария обработки;
- API-сервисы;
- сервисы распределённого анализа изображения.

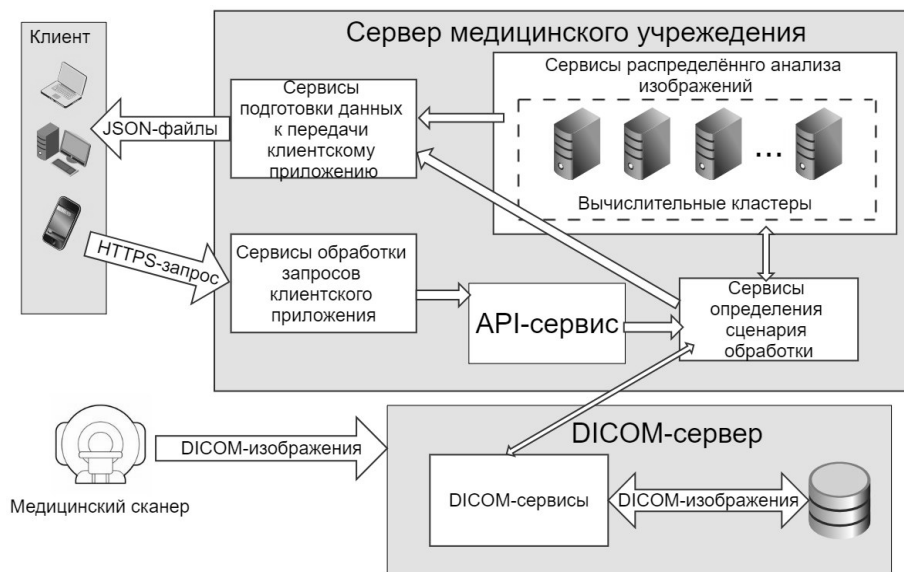


Рисунок 2 – Подробная архитектура подсистемы

Принцип работы подсистемы: с клиентского приложения по защищённому протоколу HTTPS поступает HTTP-запрос, который обрабатывается в сервисах обработки запросов. Сервисы обращаются к API-сервисам, которые подают запрос на определение сценария обработки соответственно клиентскому запросу, вследствие чего сервисы определения сценария получают от DICOM-сервера изображение и данные о нём, а также данные о загруженности ЦОД медицинского учреждения и вычислительной мощности клиентского устройства. После определения сценария задача обработки изображения выполняется в ЦОД, и клиентскому приложению приходит результат обработки, либо изображение в полном виде отправляется клиентскому приложению.

Определение сценария

Главным критерием при выборе сценария является время обработки t_o :

$$t_o = t_{mc} + t_{ck}, \quad (1)$$

где t_{ck} – время передачи данных с сервера на клиентское приложение;

t_{mc} – время обработки изображения.

Время обработки определяется по формуле:

$$t_{mc} = t_c + t_n + t_\phi, \quad (2)$$

где t_c – время считывания изображения;

t_n – время преобразования изображения определёнными методами;

t_ϕ – время формирования данных для отображения.

Время выполнения каждого этапа обработки зависит от следующих факторов:

- производительности (M) устройства, на котором обрабатывается изображение;
- размера (S) изображения;
- сложности (D) алгоритма каждого этапа обработки.

Следовательно, формула (2) имеет вид:

$$t_{mc} = \frac{M}{S} \cdot \left(\frac{1}{D_c} + \frac{1}{D_n} + \frac{1}{D_\phi} \right), \quad (3)$$

где D_c – сложность алгоритма считывания изображения;

D_n – сложность алгоритма преобразования изображения определёнными методами;

D_ϕ – сложность алгоритма формирования данных для отображения.

Главный параметр $t_{ок}$, который учитывается при принятии решения об обработке на клиентском оборудовании, определён формулой:

$$t_{ок} = t_{ckk} + t_{mck}, \quad (4)$$

где t_{ckk} – время передачи необработанных данных с сервера клиенту;

t_{mck} – время обработки на клиентском терминальном оборудовании.

Время обработки изображения на клиентском терминальном оборудовании (t_{mck}) зависит от:

- производительности (M_k) клиентского терминального оборудования;
- возможности (B) выполнения обработки ($B = 0$, если возможности нет, иначе $B = 1$);
- размера (S) изображения;
- сложности (D) алгоритма каждого этапа обработки.

Таким образом, время обработки изображения на клиентском терминальном

оборудовании может быть определено по формуле:

$$t_{мск} = \frac{B \cdot M_{к}}{S} \cdot \left(\frac{1}{D_c} + \frac{1}{D_n} + \frac{1}{D_{\phi}} \right) \quad (5)$$

Главный параметр $t_{оц}$, который определяет условие обработки изображения в ЦОД медицинского учреждения, определяется как:

$$t_{оц} = t_{скц} + t_{мсц}, \quad (6)$$

где $t_{скц}$ – время передачи результатов обработки с сервера клиенту;

$t_{мсц}$ – время обработки изображения в ЦОД медицинского учреждения.

В свою очередь, $t_{мсц}$ определено как:

$$t_{мсц} = \frac{M_{кл} \cdot K}{S} \cdot \left(\frac{1}{D_c} + \frac{1}{D_n} + \frac{1}{D_{\phi}} \right), \quad (7)$$

где K – количество кластеров, между которыми можно распараллелить обработку изображения;

$M_{кл}$ – производительность одного кластера.

Критерий, на основании которого принимается решение об обработке в ЦОД или на клиентском оборудовании, может быть сформулирован следующим образом:

– при $t_{ок} > t_{оц}$ обработку следует выполнять в ЦОД с учётом ограничений: загрузка кластеров ЦОД на момент запроса, возможность обработки изображения на клиентском терминальном оборудовании;

– при $t_{ок} < t_{оц}$ обработку следует выполнять на клиентском терминальном оборудовании.

Выводы

Развитие методов диагностики, основанных на получении и анализе медицинских изображений, а также темпы накопления этих изображений позволяют сделать вывод, что повышение быстродействия, гибкости и масштабируемости систем хранения и обработки медицинских изображений – актуальная задача. Анализ архитектур современных информационных систем показал, что масштабируемость и гибкость могут быть повышены за счёт применения принципов сервис-ориентирования. Показано, что оперативность работы и доступность системы возможно повысить за счёт автоматического определения сценария обработки на основании объёма задачи, характеристик клиентского оборудования и текущей доступности ЦОД. Сформулирован критерий выбора узла для обработки. Направлением дальнейших исследований является экспериментальное исследование применяемых методов обработки изображений, их реализаций и типов медицинских изображений с целью определения характеристик производительности

оборудования, сложности алгоритма, а также уровня загрузки сети и ЦОД, которые позволят эффективно выполнять выбор сценария обработки.

Список литературы

1. Назаренко Г. И., Гулиев Я. И., Ермаков Д. Е. Медицинские информационные системы: теория и практика / под ред. Г. И. Назаренко, Г. С. Осипова. – М.: ФИЗМАТЛИТ, 2005. – 320 с.
2. Гонсалес Р., Вудс Р. Цифровая обработка изображений / М.: Техносфера, 2005. – 1072 с.
3. Попов С. Б. Концепция распределенного хранения и параллельной обработки крупноформатных изображений / С. Б. Попов // Компьютерная оптика – 2007. – т. 31 – №4 – С. 77-85.
4. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. Ван Стеен. – СПб.: Питер, 2003. – 877 с.
5. Юдифь Гурвиц и др. Сервис-ориентированная архитектура для чайников. = Judith Hurwitz et al, Service Oriented Architecture for dummies, Wiley, 2007. – 387pp.
6. Томас Эрл, Стивен Беннетт. Управление СОА. Управление разделяемыми службами локально и в облаке. = Thomas Erl, Stephen Bennett, SOA governance. Governing shared services on-premise and in the cloud. // Prentice Hall, 2011. – 704pp.
7. Томас Эрл, Бенджамин Карлайл, СОА с REST. Принципы, шаблоны и ограничения для построения корпоративных решений на основе REST. = Thomas Erl, Benjamin Carlyle, SOA With REST. Principles, patterns and constraints for building enterprise solutions with REST. // Prentice Hall, 2012. – 624pp.
8. Евгений Алексеев, Обзор Windows Workflow Foundation на примере построения системы электронного документооборота [Электронный ресурс]. – <https://habrahabr.ru/company/luxoft/blog/181562/>
9. Амбили К.К., Настройка Windows Workflow Foundation 4.5: разработка унифицированных и масштабируемых приложений. = Ambily K. K., Customizing Windows Workflow Foundation 4.5: Unified and Scalable Application Development, Amazon Digital Services LLC, 2015. – 126 pp.
10. Симоненко А. А., Привалов М. В. Разработка веб-ориентированной подсистемы доступа к медицинским изображениям // Информатика, управляющие системы, математическое и компьютерное моделирование в рамках II форума «Инновационные перспективы Донбасса» (ИУСМКМ-2016): VII Международная научно-техническая

конференция, 26 мая 2016, г. Донецк – Донецк:
ДонНТУ, 2016. – 624 с. – С. 274-279.

Двойкин А. А., Привалов М. В., Воронова А. И. Архитектура и сценарии распределённой обработки медицинских изображений веб-ориентированной PACS. Выполнен анализ и описаны ключевые особенности систем обработки медицинских изображений, описана необходимость распределения нагрузки между клиентским приложением и сервером (центром обработки данных). Показаны основные сценарии обработки данных в такой системе и предложены критерии для их выбора.

Ключевые слова: системы обработки медицинских изображений, распределённая обработка, распределение нагрузки.

A.A. Dvoykin, M.V. Privalov, A.I.Voronova Architecture and distributed medical image processing scenarios of the Web-oriented PACS. Analysis of the medical image processing systems was performed and their key features were described. It is shown that workload balancing between client application and server (data center) is required. Main data processing scenarios and criteria for their selection were proposed.

Keywords: medical image processing systems, distributed processing, workload balancing.

*Статья поступила в редакцию 20.09.2017
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом*

УДК 519.725, 625.7, 681.3

Применение методов помехоустойчивого кодирования для компактного тестирования цифровых схем

О.Н. Дяченко, Ю.Е. Зинченко, В.О. Дяченко
Донецкий национальный технический университет
do@donntu.org

Дяченко О.Н., Зинченко Ю.Е., Дяченко В.О. Применение методов помехоустойчивого кодирования для компактного тестирования цифровых схем. Выполнен анализ эффективности компактного тестирования комбинационных схем, учитывающего структуру генератора тестовых последовательностей, анализатора тестовых реакций, построенных на основе методов циклического кодирования, и характер распределения ошибок в тестовой реакции. Предложен метод расчета сигнатур, отличающийся от известного, в выборе степенного обозначения тестовых наборов вместо двоичного. Показано, что главное достоинство аналитического вычисления сигнатур являются не столько значения компактных оценок, а вывод о сигнатурной тестируемости комбинационных схем. Этот вывод распространяется на многовходовые анализаторы тестовых реакций; для структур с посимвольным перемежением; для порождающих полиномов, как в конфигурации Галуа, так и в конфигурации Фибоначчи. Предложены рекомендации по выбору порождающих полиномов регистров сдвига с линейными обратными связями для различных вариантов компактного тестирования.

Ключевые слова: Генератор тестовых последовательностей, анализатор тестовых реакций, регистр сдвига с линейными обратными связями, порождающий полином, циклические коды.

Введение

Всемирная паутина разрастается стремительными темпами: если в 2000 году было 359 млн. пользователей сети Интернет, то в марте 2017 года количество пользователей уже 3732 млн. – это половина населения земного шара. Наша планета постепенно оказывается в своеобразном информационном “коконе”. Увеличение количества информации, которая передается, хранится и обрабатывается, приводит к требованиям обеспечения ее достоверности и надежности используемых аппаратных и программных средств. От успешного решения этих задач, с одной стороны, зависит процветание нынешней цивилизации, или, с другой стороны – ее саморазрушение, например, из-за случайного или намеренного сбоя в военных приложениях. Поэтому для устранения последствий возможных ошибок или дефектов аппаратных средств используется весь арсенал методов и средств помехоустойчивого кодирования, встроенного самотестирования [1-8].

Одним из способов повышения тестопригодности СБИС микропроцессоров, устройств на ПЛИС является применение встроенных средств контроля, реализующих методы компактного

тестирования. Метод сквозного сдвигового регистра (LSSD - level sensitive scan design) - другой широко известный способ снижения трудоемкости тестирования дискретных устройств. Метод LSSD сводит задачу тестирования к проверке нескольких регистров сдвига и комбинационных схем. Наиболее совместимым с методом LSSD, из широкого ряда методов компактного тестирования, является сигнатурный анализ, поскольку основой анализатора тестовых реакций (АТР) в этом случае является регистр сдвига с линейными обратными связями (РСЛОС). С помощью незначительных аппаратных затрат сдвиговые регистры преобразуются в РСЛОС, которые выполняют роль генераторов тестовых последовательностей (ГТП) и АТР для тестирования комбинационных схем (КС).

Реализация методов компактного тестирования ставит задачу определения достоверности результатов контроля. В работе [3] рассматриваются вопросы комплексной оценки достоверности тестирования КС при применении ГТП и АТР в виде РСЛОС, которая учитывает не только обнаруживающие способности АТР, но также структуру ГТП и характер тестовых реакций объекта диагностики. В частности, получен вывод о значительной зависимости

эффективности сигнатурного анализа от выбора того или иного сочетания порождающих полиномов РСЛОС ГТП и АТР.

Данная работа представляет собой продолжение исследований в этом направлении.

Цель статьи

Выполнить анализ эффективности компактного тестирования цифровых схем при предположении, что ГТП - РСЛОС с примитивным образующим полиномом, а АТР реализованы на основе циклических кодов. Особенностью таких анализаторов является их способность локализовать ошибки, кратность которых не превышает корректирующих способностей кодов, на основе которых они построены.

Практическая реализация самотестирования СБИС

Анализ диагностического обеспечения микропроцессорных СБИС ведущих зарубежных фирм: IBM (S/390, метод LSSD); Hewlett Packard (сигнатурный анализ); альянс компаний Apple, IBM и Motorola (Power PC); Motorola (MC 202-206); Intel Corporation (микропроцессоры 80386, Pentium Pro); Advanced Micro Devices (AMD-K6), показывает, что 5-8% площади кристалла СБИС занимают встроенные схемы тестирования, которые позволяют обнаружить практически 100% дефектов. Например, диагностическое обеспечение микропроцессора S/390 включает: ОЗУ, кэш, память, схемы их управления со встроенными схемами самотестирования; триггеры, регистровые сети, образующие в режиме тестирования сканируемый путь по методу LSSD; встроенные ГТП; встроенный АТР - многоканальный сигнатурный анализатор; порт JTAG в соответствии со стандартом IEEE 1149.1.

Методы исчерпывающего тестирования КС и сканирования позволяют вместе обнаруживать 95% неисправностей. Применение разных псевдослучайных последовательностей обеспечивает 99,9% покрытия всех неисправностей СБИС.

Аналитический метод вычислений компактных оценок

Прежде всего, рассмотрим метод аналитического расчета сигнатур, альтернативный методу, предложенному в [3].

Предположим, что ГТП и АТР реализованы в виде РСЛОС с внутренними сумматорами в цепях обратной связи с порождающими полиномами соответственно $h(X)$ и $g(X)$, причем оба полинома примитивные, а их корни связаны равенством

$$\beta = \alpha^k, m = \text{deg}h(X) = \text{deg}g(X).$$

Тестовые наборы, которые поступают на входы исследуемой КС, представляют собой ненулевые элементы поля $GF(2^m)$, являющегося расширением поля $GF(2)$ над полиномом $h(X)$. Эти элементы поля могут быть представлены в двоичном, полиномиальном и степенном обозначениях. Каждому ненулевому элементу α^k поля $GF(2^m)$ соответствует минимальный полином, причем, если минимальный полином примитивный, то его степень равна m . Если в качестве порождающего полинома РСЛОС АТР выбрать минимальный полином, соответствующий элементу α^k , то между корнями полиномов $h(X)$ и $g(X)$ будет выполнено равенство $\beta = \alpha^k$. Анализ таблицы минимальных полиномов показывает, что для любой степени $m < 5$ существует только два примитивных полинома, причем $\beta = \alpha^1$, т. е. эти полиномы являются двойственными (взаимобратными) [6]. Поэтому для примеров, иллюстрирующих метод аналитического расчета сигнатур, будем рассматривать $h(X)$ и $g(X)$ степени $m=5$.

Основное отличие предлагаемого метода расчета сигнатур от известного [3] заключается в выборе степенного обозначения тестовых наборов вместо двоичного. В этом случае значение сигнатуры для конъюнкции с рангом m может быть вычислено согласно следующему выражению: $S = M_k X^{-Ak}$, где X^A - степенное обозначение тестового набора, M - матрица для перехода от значений РСЛОС ГТП к значениям РСЛОС АТР.

Рассмотрим порядок расчета сигнатур на примерах.

Пример. Пусть $F_1 = \bar{x}_5 \bar{x}_4 x_3 x_2 x_1$, $g(X) = X^5 + X^3 + 1$, т.е. $k=-1$.

Элемент поля $GF(2^5)$ над полиномом $h(X) = X^5 + X^2 + 1$, соответствующий тестовому набору 00111, может быть представлен в полиномиальном обозначении $X^2 + X + 1$. Для перехода к степенному обозначению воспользуемся таблицей логарифмов Зеча (табл. 1). Согласно этой таблице $X^1 = X^1 + 1$. Учитывая свойства $X^{2j} = (X^j + 1)^2$ и $X^i = (X^{p^i} + 1)$, где $p=2^m-1$, определим таблицу для нечетных значений i от 1 до 15.

Таким образом, $X^2 + X + 1 = X \cdot (X + 1) + 1 = X \cdot X^{18} + 1 = X^{19} + 1 = X^{11}$. Значение сигнатуры $S(F_1) = M_{-1} X^{11}$.

Таблица 1. Логарифмы Зеча для $h(X) = X^5 + X^2 + 1$

| | | | | | | | | |
|---|----|----|---|----|----|----|----|----|
| i | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 |
| j | 18 | 29 | 2 | 22 | 16 | 19 | 14 | 24 |

Несколько замечаний по поводу построения матрицы M_k . Прежде всего, следует отметить, что вид матрицы M_k зависит не только от k , но также от начального значения РСЛОС

ГТП, которое может быть выбрано любым ненулевым.

Каждому элементу поля $\alpha^0, \alpha^1, \dots, \alpha^{m-1}$ ставится в соответствие строка матрицы M_k , которая определяется следующим образом. Для элемента α^j отыскивается значение степени j эквивалентного элемента $\alpha^j = \alpha^i$, которое нацело делится на $-k$. Значение j определяется на основании равенства $\alpha^i = \alpha^{i+pd}$, где d - любое целое число. Строка матрицы M_k представляет собой остаток от деления полинома $X^{p+j/k+(s-1)}$ на полином $g(X)$ (α^s - начальное состояние РСЛОС ГТП).

Для рассматриваемого примера при начальном состоянии РСЛОС ГТП, равном 00001, для строки, соответствующей α^0 , значение равно остатку от деления полинома $X^{31-0-1} = X^{30}$ на полином $g(X) = X^5 + X^3 + 1$.

Для начального состояния РСЛОС ГТП, равного α^0 , и $k=-1$, матрица M_{-1} представляет собой последние m состояний РСЛОС АТР.

Для начального состояния РСЛОС ГТП равного α^m и $k=-1$ матрица M_{-1} представляет собой единичную матрицу, например, при $m=5$ значение сигнатуры конъюнкции с рангом m будет равно двоичному обозначению соответствующего элемента поля, записанному в обратном порядке.

Два варианта матрицы M_{-1} будут иметь вид:

$$M_{-1} = \begin{pmatrix} 01011 \\ 10110 \\ 00101 \\ 01010 \\ 10100 \end{pmatrix}, \quad M_{-1} = \begin{pmatrix} 00001 \\ 00010 \\ 00100 \\ 01000 \\ 10000 \end{pmatrix} - \text{единичная}$$

матрица.

Итак, $S(F_1) = M_{-1}X^{11}$.

Для умножения на матрицу необходимо перейти от степенного обозначения тестового набора k двоичному (или полиномиальному). Такое преобразование можно упростить, используя заранее вычисленные значения $X^1, X^2, X^4, X^8, X^{16}$ и т.д. по модулю $h(X)$ (эффективность такого упрощения увеличивается с ростом значения m): $X^1 \bmod h(X) = X^1, X^2 \bmod h(X) = X^2, X^4 \bmod h(X) = X^4, X^8 \bmod h(X) = X^3 + X^2 + 1, X^{16} \bmod h(X) = X^4 + X^3 + X + 1$.

Произвольный элемент X_j можно представить в виде произведения полиномов со степенями степени 2: $X^{11} = X^8 \cdot X^2 \cdot X^1$. Остаток от деления X^{11} на $h(X)$ будет равен остатку от деления на $h(X)$ произведения остатков сомножителей: $(X^3 + X^2 + 1) \cdot X^2 \cdot X^1 \bmod h(X) = (X^6 + X^5 + X^3) \bmod (X^5 + X^2 + 1) = X^2 + X + 1$, или в двоичном обозначении 00111.

При начальном значении РСЛОС ГТП α^5 значение сигнатуры $S(F_1) = 11100$.

Пример. Пусть $F_2 = \bar{x}_5 \bar{x}_4 x_3 x_2$ при тех же

значениях $h(X), g(X)$ и том же начальном состоянии РСЛОС ГТП.

$$S(F_2) = M_{-1}(X^{19}) + M_{-1}(X^{19} + 1) = \\ = M_{-1}(X^{19} + X^{19} + 1) = M_{-1}(00001) = 10000.$$

Пусть $F_3 = \bar{x}_5 \bar{x}_4 x_3 x_1, F_4 = \bar{x}_4 x_3 x_2 x_1,$

$$F_5 = \bar{x}_5 \bar{x}_4 x_3.$$

$$S(F_3) = M_{-1}(X^5 + X^5 + X) = M_{-1}(X) = \\ = M_{-1}(00010) = 01000.$$

$$S(F_4) = M_{-1}(X^{11} + X^{11} + X^4) = M_{-1}(X^4) = \\ = M_{-1}(10000) = 00001.$$

$$S(F_5) = M_{-1}(X^2 + X^2 + 1 + X^2 + X + X^2 + X + 1) = M_{-1}(0) = 0.$$

Таким образом, в общем случае для $k=-1$ сигнатура конъюнкции с рангом $r=m-1$ равна произведению матрицы M_{-1} и X^i , где i - индекс отсутствующей переменной, уменьшенный на единицу; сигнатура конъюнкции с $r < m-1$ равна нулю.

Аналогично, для произвольных примитивных полиномов $h(X)$ и $g(X)$ степени m , корни которых связаны равенством $\beta = \alpha^{-3}$, для конъюнкции с рангом $r < m-2$ $S = M_{-3}(0) = 0$. Сигнатура равна нулю в следующих случаях: $k=-5, r < m-3; k=-7, r < m-4; k=-9, r < m-3; k=-11, r < m-4; k=-13, r < m-4$; для произвольного k $r < m-1-w$, где w - вес двоичной записи $-k$.

Если рассматривать полученный результат при конкретных значениях m , условие равенства сигнатуры нулю можно сформулировать иначе: $r < w[(k_0)]$, где $w[(k_0)]$ - вес двоичной записи k в обратном коде.

Таким образом, если неисправность в КС, описываемой функцией F , приводит к тестовой реакции F_n , и в представлении $F + F_n$ в виде полинома Жегалкина присутствуют только конъюнкции с рангом $r < w[(k_0)]$, то $S(F + F_n) = S(F) + S(F_n) = 0$, или $S(F) = S(F_n)$, т.е. неисправность будет необнаруженной.

Итак, число w (вес двоичной записи $-k$) представляет собой параметр, с помощью которого можно оценить эффективность сигнатурного анализа при применении в качестве ГТП и АТР РСЛОС с порождающими примитивными полиномами одинаковой степени. Параметр w принимает минимальное значение 1 при $k=-1$, и максимальное значение $m-1$ при $k=1$. Этот вывод также справедлив для многовходовых АТР и РСЛОС ГТП и АТР с альтернативной реализацией (с внешними сумматорами в цепях обратной связи).

Учитывая, что структура РСЛОС с приводимым образующим полиномом, который представляет собой произведение различных неприводимых полиномов, эквивалентна совокупности структур РСЛОС с неприводимыми полиномами - сомножителями, получают следующие выводы. Максимальная эффективность компактного

тестирования КС достигается для следующих вариантов:

- 1) АТР на основе кодов Хэмминга (локализация одиночных ошибок) - одинаковые образующие полиномы РСЛОС ГТП и АТР;
- 2) АТР на основе кодов БЧХ (локализация кратных независимых ошибок) - первый сомножитель образующего полинома РСЛОС АТР (минимальный полином, соответствующий элементу поля Галуа с наименьшей степенью j) должен быть равен примитивному образующему полиному РСЛОС ГТП. Остальные сомножители - минимальные полиномы, соответствующие элементам поля Галуа со степенями $j+2, j+4$ и т.д.
- 3) многоходовый АТР на основе кодов Бартона (локализация одиночного ошибочного вектора) - сомножитель образующего полинома РСЛОС АТР, соответствующий неприводимому полиному, должен быть равен примитивному образующему полиному РСЛОС ГТП;
- 4) многоходовый АТР на основе кодов Рида-Соломона, исправляющего одиночные ошибки (локализация одиночного ошибочного вектора) - корень первого сомножителя образующего полинома РСЛОС АТР должен быть равен корню примитивного образующего полинома РСЛОС ГТП;
- 5) многоходовый АТР на основе кодов Рида-Соломона, исправляющего ошибки кратности t (локализация t ошибочных векторов) - корень первого сомножителя образующего полинома РСЛОС АТР (элемент поля Галуа со степенью j) должен быть равен корню примитивного образующего полинома РСЛОС ГТП; корни остальных сомножителей - элементы поля Галуа со степенями $j+1, j+2, j+3$ и т.д.

Выводы

Выполнен анализ эффективности компактного тестирования с локализацией ошибок в выходной тестовой реакции проверяемой комбинационной схемы. Предложен метод аналитического расчета сигнатур для исчерпывающего тестирования комбинационных схем при реализации генератора тестовых последовательностей и анализатора тестовых реакций на основе регистров сдвига с линейными обратными связями. Рассматриваются анализаторы тестовых реакций, построенные на основе циклических кодов Хэмминга, Боуза-Чоудхури-Хоквингема, Бартона, Рида-Соломона, позволяющие локализовать ошибки в тестовой реакции. Получена оценка меры эффективности компактного тестирования комбинационных схем, на основе которой предложены рекомендации по выбору порождающих полиномов регистров сдвига с линейными обратными связями для

различных вариантов компактного тестирования с локализацией ошибок. Полученные результаты могут найти применение для встроенного самотестирования или внешнего тестового оборудования средств вычислительной техники.

Литература

1. Richard E. Blahut. Algebraic Codes for Data Transmission/ Cambridge University Press, 2012. – 498 p.
2. Дяченко О.Н., Дяченко В.О. Альтернативный метод укорачивания циклических кодов // Электронные информационные системы. 2017. №1 (12). С. 94–100.
3. Ярмолик В.Н., Калоша Е.П. Эффективность сигнатурного анализа в самотестирующихся СБИС // Электрон. моделирование. 1992. 14, №3. С.51-56.
4. Дяченко О.Н. Анализ сигнатурной тестируемости комбинационных схем // Автоматика и вычислительная техника. - 1990. - №5. С.85-89.
5. Ярмолик В.Н., Калоша Е.П. Эффективность сигнатурного анализа в самотестирующихся СБИС // Электрон. моделирование. 1992. 14, №3. С.51-56.
6. Ершов А.Н., Петров С. В., Пятошин Ю.П., Коханько Д. В., Зяблов В.В. и др. Улучшение радиационной стойкости памяти с помощью помехоустойчивых кодов // Ракетно-космическое приборостроение и информационные системы. 2014, том 1, выпуск 4, С. 42–49.
7. Дяченко В.О., Дяченко О.Н. Циклическое кодирование цифровой информации на основе двойственных полиномов // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике: материалы II Международной научно-практической конференции (Азов, 19 мая 2015 г.) – Ростов н/Д, ДГТУ, 2015. С. 71-76.
8. Дяченко В.О., Зинченко Ю.Е., Дяченко О.Н. Исследование способов проектирования кодов Рида-Соломона // Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ-2014): V Всеукраїнська науково-технічна конференція студентів, аспірантів та молодих вчених, 22-23 квітня 2014 р., м. Донецьк : зб. доп. / Донец. націонал. техн. ун-т; редкол. В.А.Світлична. – Донецьк: ДонНТУ, 2014. – в 2 тт. – т.2. – С. 72-78.

9. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. – М.: Мир, 1976. – 595 с.: ил.

Дяченко О.Н., Зинченко Ю.Е., Дяченко В.О. Применение методов помехоустойчивого кодирования для компактного тестирования цифровых схем. Выполнен анализ эффективности компактного тестирования комбинационных схем, учитывающего структуру генератора тестовых последовательностей, анализатора тестовых реакций, построенных на основе методов циклического кодирования, и характер распределения ошибок в тестовой реакции. Предложен метод расчета сигнатур, отличающийся от известного, в выборе степенного обозначения тестовых наборов вместо двоичного. Показано, что главное достоинство аналитического вычисления сигнатур являются не столько значения компактных оценок, а вывод о сигнатурной тестируемости комбинационных схем. Этот вывод распространяется на многоходовые анализаторы тестовых реакций; для структур с посимвольным перемежением; для порождающих полиномов, как в конфигурации Галуа, так и в конфигурации Фибоначчи. Предложены рекомендации по выбору порождающих полиномов регистров сдвига с линейными обратными связями для различных вариантов компактного тестирования.

Ключевые слова: Генератор тестовых последовательностей, анализатор тестовых реакций, порождающий полином, циклические коды.

Dyachenko O. N., Zinchenko Y.E., Dyachenko V. O. Application of methods of antinoise coding for compact testing of digital circuits. The efficiency of compact testing of combinational circuits that takes into account the structure of the generator of test sequence, analyzer of test reactions based on the methods of cyclic encoding and the distribution of errors in a test response, under study. The calculation method signatures differ from the well-known in the selection of a power notation of test cases instead of binary is proposed. It is shown that the main advantage of the analytical evaluation of the signatures are not so much values compact evaluations, and the conclusion of signature testability of combinational circuits. This conclusion is extended to multi-analyzers test reactions; for structures with symbol-by-symbol interleaving; for the generator polynomials, in Galois configuration and the configuration of the Fibonacci. Recommendations on the choice of the generator polynomials of linear feedback shift register for different variants of compact testing are proposed.

Keywords: Generator of test sequences, analyzer of test reactions, linear feedback shift register, generator polynomial, cyclic codes.

Статья поступила в редакцию 20.09.2017
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам

Ю. Е. Зинченко, В. С. Ленч, Т. А. Зинченко, В.Н. Павлыш
Донецкий национальный технический университет
zinchenko.tatyana@gmail.com

Ю. Е. Зинченко, В. С. Ленч, Т. А. Зинченко, В.Н. Павлыш Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам. Целью работы является разработка и реализация метода имитации периферийных устройств, подключаемых к FPGA, а также разработка личного кабинета пользователя на Веб-сайте виртуальной лаборатории, и обеспечение возможности удаленного программирования FPGA-комплекса. Данная разработка необходима для упрощения доступа к дорогостоящим FPGA-комплексам студентам и инженерам, работающим в сфере компьютерной инженерии на базе FPGA. Объектом исследования являются методы имитации физических устройств, подключаемых к ПЛИС типа FPGA, которые позволяют организовать удаленный доступ к отладочным FPGA-комплексам. В работе был выполнен анализ существующих методов имитации различных физических устройств для обеспечения удаленной отладки FPGA-проектов. Были исследованы способы организации удаленного доступа к программируемым комплексам. На основании результатов анализа был разработан метод имитации периферийных устройств, подключаемых к FPGA-комплексу, с возможностью удаленного управления.

Ключевые слова: ПЛИС, FPGA, VHDL, РЭА, ЭВА, FPGA Remote Control, эмуляция, удаленный доступ, Internet

1 Введение

Сегодня в мире в целом и в странах СНГ в частности широко развиваются FPGA-технологии как одна из альтернатив проектирования аппаратных средств радиоэлектронной (РЭА) и электронно-вычислительной (ЭВА) аппаратуры. Основой этих технологий являются программируемые логические интегральные схемы (ПЛИС), и прежде всего их разновидность – FPGA (Field Programmable Gate Arrays), и языков описания (программирования) аппаратуры, таких как VHDL и Verilog[1]. Процесс проектирования на базе FPGA-технологий на этапе прототипирования обычно предусматривает использование FPGA-комплексов, однако это сдерживается относительной их дороговизной, широкой номенклатурой и необходимостью постоянно обновлять парк таких комплексов. Последнее особенно чувствительно для вузов и не только стран СНГ, но для западного региона, где доступ к ограниченному составу FPGA-комплексов нужно обеспечивать широкому кругу студентов и аспирантов. Поэтому разработка систем дистанционного доступа широкого круга специалистов и прежде всего студентов к FPGA-комплексам является, безусловно, задачей актуальной. Основной проблемой систем дистанционного

доступа к РЭА и ЭВА является обеспечение возможности имитации периферийных устройств, чтобы создавать эффект присутствия пользователя вблизи аппаратуры и как бы воспроизводить манипуляции органами управления. К настоящему времени известно уже множество таких систем, как к FPGA-комплексам, так и к другим электронным комплексам на базе микропроцессоров и микроконтроллеров[2-8]. В большинстве таких систем возможность имитации либо вообще не внедрена, либо предусматривает использование дополнительных программно-аппаратных средств, что резко усложняет, удорожает и снижает гибкость дистанционного доступа. В данной работе предлагается универсальный подход организации системы дистанционного доступа к FPGA-комплексам на базе сети Интернет с возможностью имитации периферийных устройств без использования дополнительного оборудования. Подход базируется на погружении специальных имитаторов в FPGA-кристалл в виде IP-ядер, разработанных на VHDL или Verilog, вместе с тестируемым проектом, которые способны воспринимать внешние манипуляционные команды от дистанционно удаленного оператора-отладчика проекта (инженера,

студента и т.п.). Статья является дальнейшим развитием работ [9-11], проводимых лабораторией ДонНТУ «FPGA-технологии проектирования и диагностика КС», в указанной области исследования [12].

1. Разработка метода имитации периферийных устройств для FPGA-комплекса

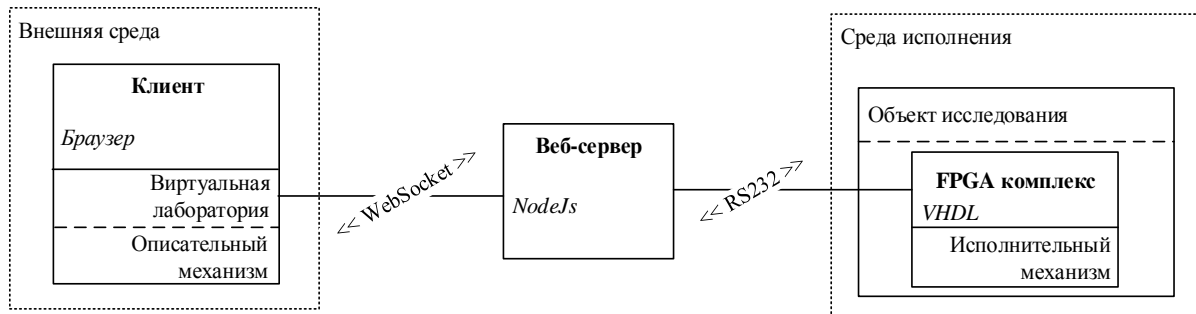


Рисунок 1 – Общая структура метода имитации в виртуальной FPGA-лаборатории

Как видно из структуры, клиентская часть находится во внешней среде, это означает, что она взаимодействует с пользователем посредством доступных устройств ввода-вывода. Также на схеме показано, что виртуальная лаборатория использует интернет-браузер для взаимодействия с веб-сервером. Такой выбор связан с отсутствием установки дополнительного программного обеспечения на персональный компьютер (ПК) пользователя, а интернет-браузер на сегодняшний день является неотъемлемой частью ПК, из-за большой популярности сети Интернет. Но такой выбор вносит свои ограничения на использование подключенного периферийного оборудования к ПК, так как это обусловлено политикой безопасности разработки интернет-браузеров.

Исходя из выбранного метода логика работы имитаторов описывается в клиентской части виртуальной лаборатории, это значит, что данные приходят от объекта исследования, а решение об использовании данных принимается на странице виртуальной лаборатории. Такой подход является менее безопасным с точки зрения использования данных, но позволяет снизить нагрузку и расход ресурсов FPGA-комплекса.

В качестве веб-сервера предпочтительно использовать серверное программное обеспечение, предназначенное для создания веб-приложений реального времени, как, например, Node.js. В сервере Node.js реализована поддержка

Выбранный метод, в связи с общей схемой реализации виртуальных лабораторий с дистанционным доступом, представляет собой общую структуру метода имитации периферийных устройств в виртуальной FPGA-лаборатории, представленной на рисунке 1.

протокола WebSocket, который позволяет передавать данные от сервера к клиенту без дополнительных запросов, что обеспечивает обмен данными в реальном времени. Из схемы на рисунке 1 видно, что серверная часть служит звеном взаимодействия между виртуальной лабораторией и FPGA-комплексом. Для связи FPGA-комплекса с сервером необходимо использовать какой-либо быстродействующий интерфейс передачи данных, как, например, RS232 или USB.

В данном случае объектом исследования и средой исполнения эмуляторов периферийных устройств является FPGA-комплекс. Также в нем содержится исполнительный механизм эмуляторов периферийных устройств, это означает, что в FPGA-комплексе расположена их аппаратная составляющая. Для того, чтобы пользователи могли разрабатывать свои проекты с использованием эмуляторов периферийных устройств, необходимо дополнять свои проекты вспомогательной заготовкой, где реализован исполнительный механизм эмуляторов устройств, а также обмен данными с сервером. Также возможно дополнять автоматически проект на этапе синтеза пользовательского устройства в FPGA, но для этого необходимо разработать дополнительное программное обеспечение для синтеза устройств.

Таким образом, структура проекта в FPGA должна иметь следующий вид, как показано на рисунке 2.

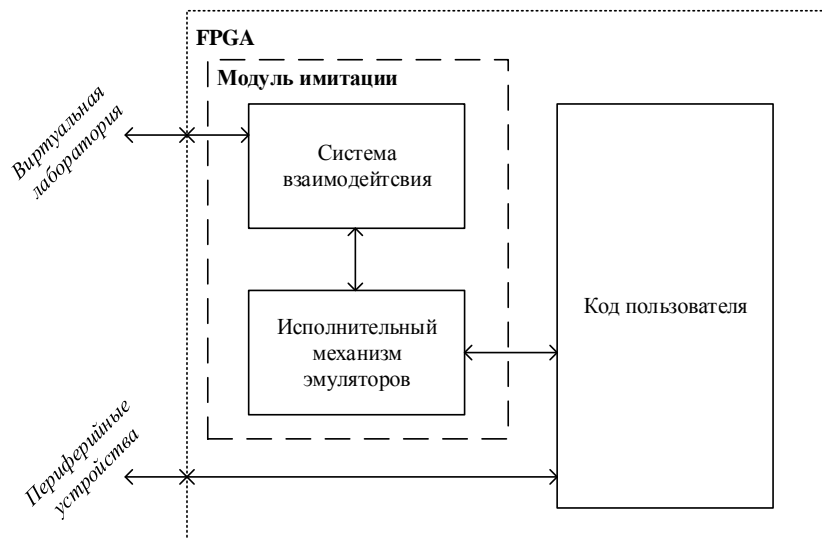


Рисунок 2 – Схема организации FPGA-проекта с эмуляторами периферийных устройств

На рисунке 2 видно, что код пользователя может использовать, как эмуляторы периферийных устройств, так и физически подключенные периферийные устройства. Такая организация связана с тем, чтобы разработчик имел возможность с минимальными трудозатратами изменять свой проект для использования, как в системе виртуальной лаборатории, так и при непосредственном доступе к периферийному оборудованию FPGA-комплекса.

Так как подключение к FPGA-комплексу организуется единым каналом связи, то необходимо разрабатывать систему взаимодействия сервера с имитаторами на плате. Задача системы взаимодействия принять данные с виртуальной лаборатории и перенаправить тому имитатору устройства, которому эти данные предназначены. Таким же образом обеспечить прием данных от имитаторов и передать их в виртуальную лабораторию.

Исполнительный механизм имитаторов представляет собой множество эмуляторов периферийных устройств, которые связаны с пользовательским кодом на уровне физических сигналов по протоколам связи, которые используют эти устройства. А с системой взаимодействия в том виде, который удобен для передачи по выбранному каналу данных, например, в виде цифровых байт данных.

Таким образом, выбранная методика имитации периферийных устройств в виртуальной FPGA-лаборатории с дистанционным доступом подразумевает под собой работу в несколько этапов. Имитатор ввода данных, в виртуальной лаборатории,

принимает данные от пользователя и обрабатывает их в соответствии с логикой работы устройства. Далее обработанные данные должны быть переданы на веб-сервер, который в свою очередь пересылает их в FPGA-комплекс по быстродействующему каналу связи. В FPGA данные должны быть приняты системой взаимодействия, которая идентифицирует целевой имитатор устройства, и пересылает ему данные. Имитатор устройства должен получить данные, преобразовать их в соответствии с протоколом работы соответствующего устройства, и переслать их в проект пользователя, который реализует поставленную задачу. В случае вывода данных необходимо произвести обратный процесс.

2. Разработка структуры взаимодействия виртуальной лаборатории с FPGA-комплексом

В общем случае виртуальная лаборатория обменивается данными с веб-сервером по сети Интернет, а сервер обменивается данными с FPGA-комплексом по физическому каналу связи. Для связи сервера с FPGA-комплексом предлагается использовать протокол передачи данных RS232, из-за простоты работы и широкой распространенности. Таким образом, следует учитывать, что протокол RS232 является последовательным, и использует две отдельные линии связи: для приема и для передачи. На рисунке 3 приведена общая схема системы взаимодействия между сервером и имитаторами в FPGA-проекте.

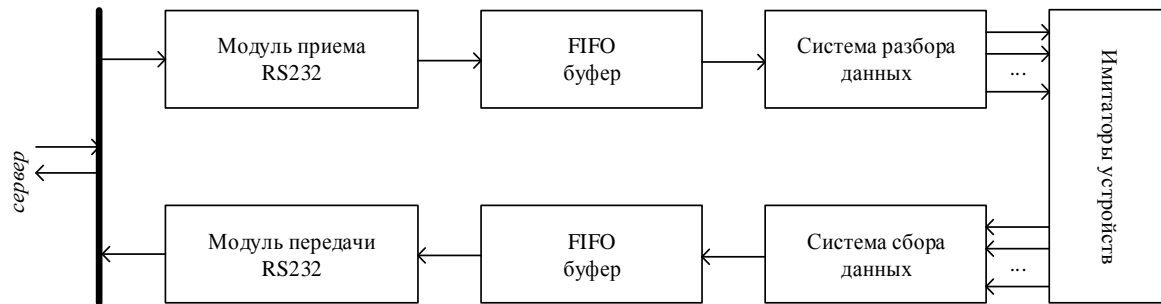


Рисунок 3 – Общая схема взаимодействия сервера с FPGA-комплексом

Такая схема позволяет с минимальными трудозатратами изменить способ обмена данными между сервером и FPGA-комплексом. Достаточно изменить модули передачи и приема в соответствии с выбранным интерфейсом связи, и схема продолжит свою работу в обычном режиме.

На схеме представлены модули приема и передачи RS232 они обеспечивают обмен данными по протоколу RS232 между FPGA-комплексом и сервером. Задачей этих модулей является обеспечение синхронизации передачи данных, а также преобразование данных из сигналов протокола связи в байты и наоборот.

Для увеличения стабильности обмена данными используются FIFO-буферы. При приеме данных может возникнуть ситуация, когда имитаторы еще не завершили прием предыдущей посылки, но сервер уже передает следующую порцию данных. В таком случае в

буфере будут накапливаться данные, и по мере работы имитаторов освобождаются. Таким же образом работает и буфер при передаче данных на сервер. Если от имитаторов устройств поступает много данных, но модуль передачи не успевает их переслать, тогда данные накапливаются в буфере и постепенно передаются на сервер.

Система разбора данных выполняет две функции: разбирает принятый пакет данных и передает данные имитатору устройства. Так как есть необходимость идентифицировать устройства, то необходимо разработать структуру пакета обмена данными. Эта структура должна содержать разделитель пакетов, идентификационный код устройства, данные и признак окончания пакета. Таким образом, была разработана структура пакета данных, представленная на рисунке 4.

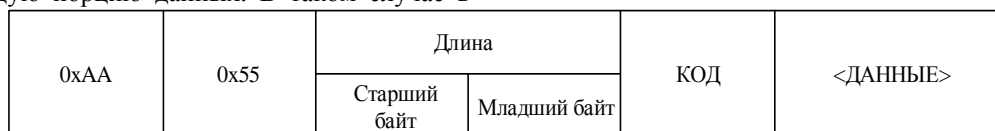


Рисунок 4 – Структура пакета данных между сервером и FPGA-комплексом

В представленной структуре двухбайтная подпись 0xAA55 является идентификатором начала пакета данных. Далее идут два байта длины данных, это поле определяет, сколько байт данных передается в пакете, и служит признаком окончания пакета. Поле «код» имеет длину один байт и представляет собой идентификационный код устройства. Поле «<данные>» содержит в себе количество байт, указанных в поле «длина». Такая структура пакета данных должна обеспечивать целостную передачу и однозначную пересылку данных между описательным и исполнительным механизмами эмуляторов периферийных устройств.

Таким же образом система сбора данных получает данные от имитаторов устройств, готовых к передаче, и формирует пакет для передачи на сервер.

Сервер в свою очередь должен обеспечивать прием и разбор пакетов данных от FPGA-комплекса и передачу их на веб-страницу виртуальной лаборатории. А также прием данных с веб-страницы виртуальной лаборатории, формирование пакета данных и передачу его в FPGA-комплекс.

На рисунке 5 приведена диаграмма программной организации имитаторов в виртуальной лаборатории.

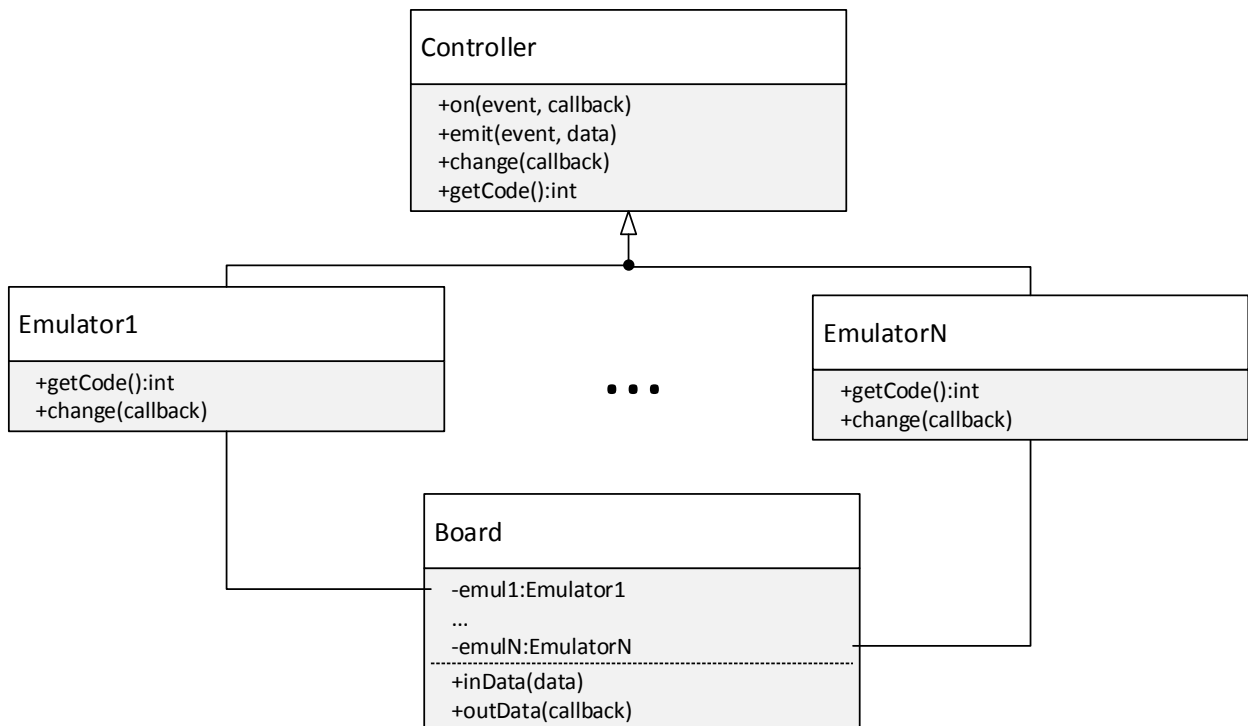


Рисунок 5 – Диаграмма программной организации имитаторов в виртуальной лаборатории

В общем случае каждый имитатор в виртуальной лаборатории должен расширять класс *Controller* и реализовывать методы *getCode()* и *change()*. Класс контроллер должен содержать в себе реализации методов *on()* и *emit()*, которые предназначены для организации событийной модели имитаторов. Метод *on(event, callback)* должен принимать на вход название события и функцию обработчик, которая будет вызываться при наступлении события, например, *on('change', function(data) { /* some actions */ })*. А метод *emit(event, data)* должен вызывать все обработчики, которые имеет событие *event*, и передавать им данные *data* в качестве параметра.

Метод *getCode()* должен возвращать целочисленный код имитатора устройства, который служит идентификатором связи с имитатором в FPGA-комплексе. Метод *change(callback)* должен представлять обертку над методом *on()* с типом события «change».

Экземпляры имитаторов должны находится в классе *Board*, который содержит два метода: *inData()* и *outData()*. Метод *inData(data)* должен вызываться главным приложением, которое принимает данные от сервера, и в качестве параметра ему передаются данные, которые содержат код, размер и сами данные. В зависимости от кода этот метод должен вызывать обработку соответствующего имитатора устройства. Метод *outData(callback)*

принимает на вход функцию обработчик, которая пересылается в метод *change()* всем экземплярам имитаторов. Обработчик, который передается в метод *outData()* должен содержать в себе обработку передачи данных на сервер. Данные должны передаваться в качестве параметра к функции обработчика, и должны иметь общую структуру для всех имитаторов устройств.

Каждый класс эмулятора должен работать с пользовательским интерфейсом веб-страницы виртуальной лаборатории и взаимодействовать с пользователем посредством различных элементов управления. Эмуляторы должны содержать в себе логику работы своего устройства и передавать данные методом *emit()*.

3. Структура систем разбора и сбора данных в FPGA-комплексе

Важнейшими частями в, разрабатываемой, системе имитации периферийных устройств являются системы разбора и сбора, пакетов данных в FPGA-комплексе. Эта система служит прикладным протоколом распознавания данных, принятых по физическому каналу связи.

На рисунке 6 приведена диаграмма состояний модуля разбора пакета данных.

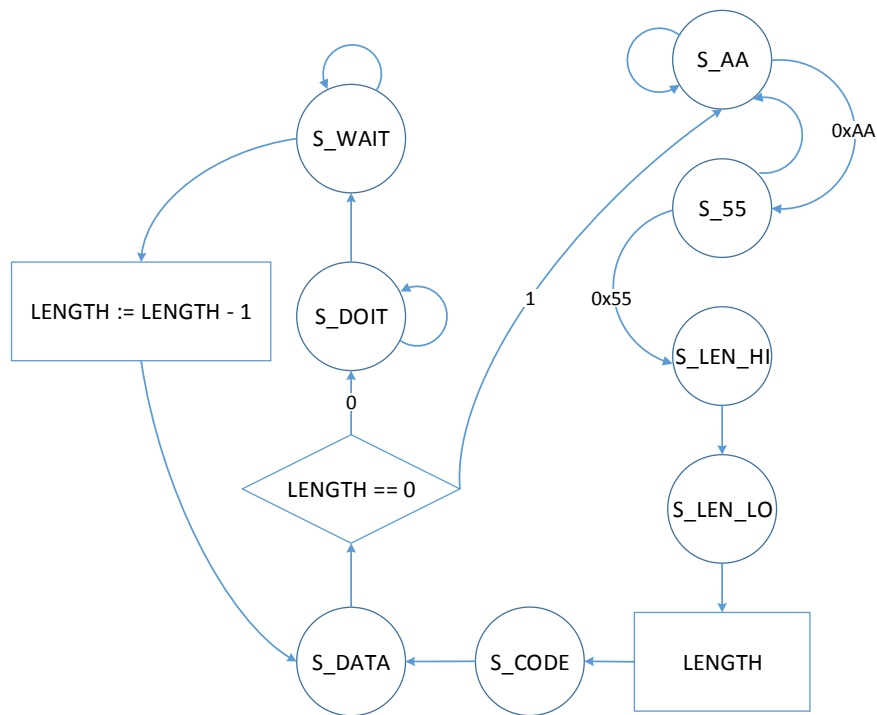


Рисунок 6 – Диаграмма состояний разбора пакета данных

Входным значением для системы разбора пакета данных является принятый байт. Первоначальным состоянием является состояние S_AA, это состояние, в котором ожидается первый байт стартовой сигнатуры пакета данных – 0xAA. Если приходит первый байт пакета данных, то состояние изменяется на S_55, которое ожидает второй байт сигнатуры – 0x55. Если двухбайтная сигнатура принята, то это означает, что далее идет корректная последовательность данных. Вначале принимается два байта длины данных, которые сохраняются в памяти. Далее принимается код имитатора устройства, который является адресом для целевого имитатора. После чего начинается пересылка данных имитатору устройства.

При пересылке данных имитатору проверяется значение длины данных. Если длина равняется нулю, то состояние переводится к начальному (S_AA). Если длина данных еще не обнулилась, то ожидается готовность приема данных имитатором устройства. Если имитатор готов к приему данных, то ему передается байт данных и состояние переводится в S_WAIT, где ожидается успешный прием имитатором устройства и после чего уменьшается на единицу длина данных. Далее состояние переводится в S_DATA, и эти операции повторяются до тех пор, пока имитатору не будут переданы все данные от сервера. Такая диаграмма разбора данных должна позволить реализовать систему

разбора пакета данных, как в FPGA-комплексе, так и на сервере.

Диаграмма состояний модуля сбора пакета данных от имитаторов периферийных устройств представлена на рисунке 7.

Схема сбора пакета данных похожа на схему разбора, только первым состоянием здесь является S_WAIT, в котором ожидается поступление данных от устройства. Когда на каком-либо имитаторе появляются данные, то перед переводом в состояние S_DOIT сохраняется его код устройства и длина данных. Состояние S_DOIT является переходным состоянием от состояния ожидания и начала передачи данных. После состояния S_DOIT каждое последующее состояние представляет собой пересылку байт, соответствующих структуре пакета данных. Так состояния S_LEN_HI и S_LEN_LO передают старшую и младшую части длины данных, а состояние S_CODE передает код имитатора устройства. Состояние S_DATA может быть расширено дополнительной обработкой, в зависимости от реализации имитатора периферийного устройства. После окончательной передачи пакета данных состояние, снова, должно установиться в S_WAIT.

Представленные на рисунках 6 и 7 диаграммы работы с пакетами данных позволяют реализовать обмен данными между зависимыми частями имитаторов устройств в виртуальной

лаборатории и FPGA-комплексе, который не зависит от физического канала связи.

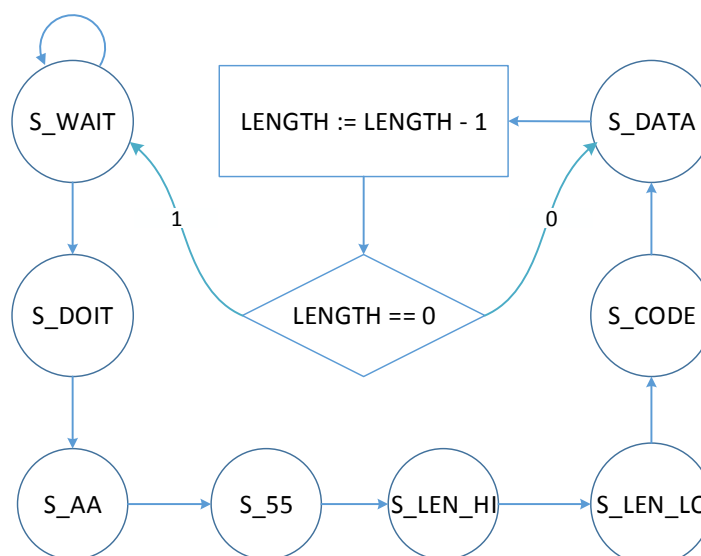


Рисунок 7 – Диаграмма состояний сбора пакета данных от имитаторов устройств

4. Общая структура исполнительного механизма имитатора устройства

Каких-либо жестких требований к реализации имитаторов периферийных устройств в FPGA нет. Но обязательно должны присутствовать, так или иначе, входные и выходные данные. На рисунке 8 представлена общая структура имитатора периферийного устройства. Пунктирной линией выделены входы и выходы, которые могут отсутствовать, в зависимости от реализуемого устройства.

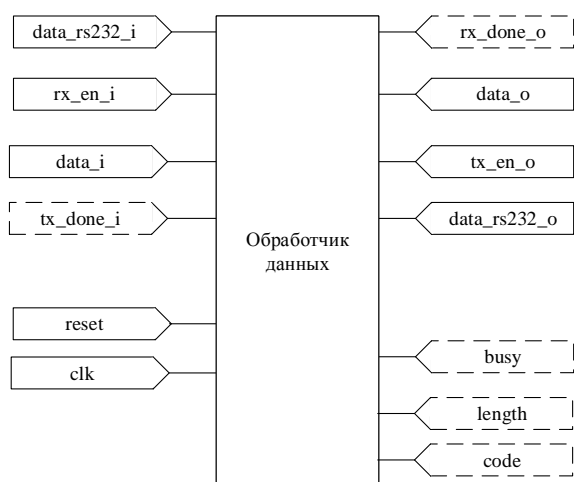


Рисунок 8 – Общая структура исполнительного механизма имитатора устройства в FPGA

Входы «*reset*» и «*clk*» являются обязательными для каждого устройства. Вход «*reset*» подключается к общей линии сброса, по которой происходит инициализация всей системы проекта. Линия «*clk*» является общей линией синхронизации, и несет в себе тактовый сигнал с частотой FPGA-комплекса. Выходы «*length*» и «*code*» должны присутствовать, так или иначе. Они не обязательно должны быть представлены, как выходы имитатора, а могут существовать, как константные значения и связаны с символьным наименованием имитатора устройства в системе. Значение «*code*» должно представлять собой идентификатор устройства, а значение «*length*» – количество выходных данных имитатора, передаваемых в виртуальную лабораторию.

Выход «*busy*» также не является обязательным, и предназначен для оповещения системы о том, что обработчик имитатора устройства, в данный момент, занят. Вход «*data_rs232_i*» предназначен для ввода данных, которые приходят от сервера, а вход «*rx_en_i*» должен оповещать имитатор о наличии новых данных на входе, готовых к приему. Об окончании приема порции данных может оповещать выход «*rx_done_o*». После обработки входных данных от лаборатории на выход «*data_o*» должны передаваться данные, в физических сигналах устройства, в пользовательский проект, где они должны использоваться в соответствии с заданной логикой.

На вход «*data_i*» поступают данные из пользовательского проекта в виде физических сигналов устройства. После их обработки на выход «*data_rs232_o*» передаются эти данные, готовые к передаче в виртуальную лабораторию. О наличии данных на выходе свидетельствует линия «*tx_en_o*». А линия «*tx_done_i*» оповещает имитатор устройства об успешной передаче на сервер.

Информационные сигналы по большей части служат средствами синхронизации при передаче данных между системами сбора и разбора данных и эмуляторами периферийных устройств. Такая схема позволит разработать исполнительный механизм имитатора периферийного устройства и связать его с описательным механизмом в виртуальной лаборатории через систему взаимодействия.

Заключение

В данной работе предложен подход имитации периферийных устройств в составе виртуальной FPGA-лаборатории с дистанционным доступом, не требующий дополнительных аппаратных комплексов, что выгодно отличает его от указанных аналогов. Разработаны структуры дистанционного взаимодействия виртуальной FPGA-лаборатории с FPGA-комплексом, а также структуры имитаторов для различных периферийных устройств, а также структуры имитаторов для различных периферийных устройств. На базе FPGA-лаборатории ДонНТУ [12] разработана экспериментальная система дистанционного доступа к FPGA-комплексам фирмы Xilinx. Проведенные эксперименты на FPGA-комплексе Spartan-3E показали положительные результаты. В настоящее время проводятся дальнейшие испытания разработанной системы.

Литература

1. Зинченко Ю., Калашников В., Хайдук С. и др. FPGA-технологии проектирования и диагностика компьютерных систем / Сб. научн. тр. VI Межд. научн.- практ. конф. «Современные информационные технологии и ИТ-образование». - Москва: МГУ, 2011. - Т. 1. 787 С. 422-429.
2. Fotopoulos V. Remote FPGA Laboratory Course Development based on an Open Multimodal Laboratory Facility / V. Fotopoulos, A. Fanariotis, T. Orphanoudakis, A. Skodras // [Электронный ресурс]: материалы 19-й всегреческой конф. по информатике (PCI 2015), Афины, Греция, 1-3 октября 2015. — Режим доступа: https://www.researchgate.net/publication/281437451_Remote_FPGA_Laboratory_Course_Development_based_on_an_Open_Multimodal_Laboratory_Facility

281437451 Remote FPGA Laboratory Course Development based on an Open Multimodal Laboratory Facility

3. Шауэрман А.А., Лаборатория Электронных Средств Обучения (ЛЭСО) СибГУТИ / А.А. Шауэрман, А.А. Подберезный [Электронный ресурс]. — Режим доступа: <http://www.labfor.ru/>

4. Reichenbach M. A New Virtual Hardware Laboratory for Remote FPGA Experiments on Real Hardware // M. Reichenbach, M. Schmidt, B. Pfundt, D. Fey. [Электронный ресурс]. — Режим доступа: <http://iuskm.donntu.org/electronic/iuskmm2017.pdf>

5. Soares J. A Remote FPGA Laboratory for Digital Design Students // Joao Soares, Jorge Lobo. ISR – Institute of Systems and Robotics DEEC, FCT, University of Coimbra [Электронный ресурс]. — Режим доступа: <http://ap.isr.uc.pt/archive/jlobo-REC2011.PDF>

6. Karthik S. Remote FIELD-Programmable Gate Array (FPGA) Lab // S. Karthik, P. Shreya, Srihari P, N.M. Viswanath. International Journal of Research in Engineering and Technology. Volume: 03 Issue: 04. Apr-2014.

7. Morgan F. Remote FPGA Lab With Interactive Control and Visualization Interface // F. Morgan, S. Cawley, M. Kane, A. Coffey, F. Callaly. Irish Signals & Systems Conference and China-Ireland International Conference on Information and Communications Technologies, Limerick, Ireland, pp. 221-226, Jun 2014 [Электронный ресурс]. — Режим доступа: <http://ieeexplore.ieee.org/document/6044871/>

8. Morgan F. Remote FPGA Lab for Enhancing Learning of Digital Systems / F. Morgan, S. Cawley, D. Newell. ACM Transactions on Reconfigurable Technology and Systems (TRETS), pp. 496-499, Oct 2012 [Электронный ресурс]. — Режим доступа: <http://ieeexplore.ieee.org/abstract/document/598152>

9. В.С. Ленч, В.А. Гень, Ю.Е. Зинченко, А.В. Коваль. Исследование организации удаленного доступа к FPGA-лаборатории // Компьютерная и программная инженерия – 2015 [Электронный ресурс]: материалы межд. научно-техн. конф. студентов, аспирантов и молодых ученых, г. Донецк, 15-16 декабря 2015г., с. 43. — Режим доступа: <http://cs.conf.donntu.org/docs/sbornik2015.pdf>

10. В.С. Ленч, Ю.Е. Зинченко, А.В. Коваль, Т.А. Зинченко Исследование методов создания эмуляторов периферийных устройств для удаленной отладки проектов в FPGA-лаборатории // Программная инженерия: методы и технологии разработки информационно-вычислительных систем (ПИИВС-2016) [Электронный ресурс]: материалы I научно-

практ. конф., г. Донецк, 16-17 ноября 2016 г. с. 221. — Режим доступа: <http://www.pi.conf.donntu.org/ПЛИВС-2016%20сборник%20докладов%20студ%20секции.pdf>

11. А.Б. Азаров, Д.А. Зубков, В.С. Константинов, Ю.Е. Зинченко, А.В. Коваль. Эмуляция периферийных устройств для удаленного доступа к FPGA-комплексам // Информатика, управляющие системы, математическое и компьютерное моделирование в рамках III форума «Инновационные

перспективы Донбасса» (ИУСМКМ - 2017) [Электронный ресурс]: материалы VIII Межд. научно-техн. конф., г. Донецк, 26 мая 2016 г. с. 686. — Режим доступа: <http://iuskm.donntu.org/electronic/iuskmm2017.pdf>

12. Зинченко Ю.Е., Калашников В.И., Дяченко О.Н., и др.. Современные проекты FPGA-лаборатории ДонНТУ / Материалы межд. научно-практ. конф. «Инновационные перспективы Донбасса», том 5. «Компьютерные науки и технологии, г. Донецк, 20-22 мая 2015 г. – Донецк: ДонНТУ, 2015. С 60-66

Ю.Е. Зинченко, В.С. Ленч, Т.А. Зинченко, В.Н. Павлыш *Имитация периферийных устройств в системе дистанционного доступа к FPGA-комплексам. Целью работы является разработка и реализация метода имитации периферийных устройств, подключаемых к FPGA, а также разработка личного кабинета пользователя на Веб-сайте виртуальной лаборатории, и обеспечение возможности удаленного программирования FPGA-комплекса. Данная разработка необходима для упрощения доступа к дорогостоящим FPGA-комплексам студентам и инженерам, работающим в сфере компьютерной инженерии на базе FPGA. Объектом исследования являются методы имитации физических устройств, подключаемых к ПЛИС типа FPGA, которые позволяют организовать удаленный доступ к отладочным FPGA-комплексам. В работе был выполнен анализ существующих методов имитации различных физических устройств для обеспечения удаленной отладки FPGA-проектов. Были исследованы способы организации удаленного доступа к программируемым комплексам. На основании результатов анализа был разработан метод имитации периферийных устройств, подключаемых к FPGA-комплексу, с возможностью удаленного управления.*

Ключевые слова: ПЛИС, FPGA, VHDL, PЭА, ЭВА, FPGA Remote Control, эмуляция, удаленный доступ, Internet

Y. E. Zinchenko, V.S. Lench, T.A. Zinchenko, V.N. Pavlysh. *Simulation of peripheral devices in the system of remote access to FPGA-complexes. The aim of the work is to develop and implement a method for simulating peripheral devices connected to the FPGA, as well as developing a personal user cabinet on the Virtual Laboratory Web site, and providing the possibility of remote programming of the FPGA-complex. This development is necessary to simplify access to expensive FPGA-complexes for students and engineers working in the field of computer engineering based on FPGA. The object of the study are methods for simulating physical devices connected to FPGA type FPGAs that allow remote access to FPGA-based debugging systems. In work the analysis of existing methods of imitation of various physical devices for maintenance of remote debugging of FPGA-projects has been executed. The methods of remote access to programmable complexes were explored. Based on the results of the analysis, a method was developed for simulating peripheral devices connected to an FPGA-complex, with the possibility of remote control.*

Keywords: FPGA, FPGA, VHDL, REA, EVA, FPGA Remote Control, emulation, remote access, Internet

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

Проектирование компьютерных компонентов для реализации операций тетралогии

Иваница С. В.

Донецкий национальный технический университет
ivanitsa-serg@rambler.ru

Иваница С. В. Проектирование компьютерных компонентов для реализации операций тетралогии. Выполнен синтез элементов тетралогии, выполняющих тетрафункции над входными сигналами (операндами, данными). Предложены модели компьютерных компонентов, выполненных в среде Active-HDL. Определены зависимости между унарными сдвиговыми и инверсными логическими операциями тетралогии с возможностью универсального использования синтезированных элементов.

Ключевые слова: тетракод, тетрит, тетралогика, логические элементы, постбинарный компьютеринг.

Введение

В 1996 году в работе [1] впервые изложена концепция тетралогии и тетракодов как с теоретической точки зрения, так и с точки зрения возможных практических приложений. Позже, в работах [2, 3] было выделено двумерное логическое пространство с определенными в нем состояниями тетралогии. Использование абстрактной модели двумерного логического пространства позволило определить унарные и бинарные логические операции для тетралогии (тетралогические операции) [3]. Полученные результаты тетралогических операций были подтверждены при представлении и взаимодействии возможных состояний тетралогии с применением аксиоматического аппарата теории множеств [4].

Целью данной работы является

1. Синтез компонентов, выполняющих определенные в [3, 4] тетралогические операции.
2. Проектирование и моделирование синтезированных компонентов в среде Active-HDL.

Разработка и проектирование базовых элементов тетралогии

Аналогично понятию логических элементов двоичной логики, справедливо нижеследующее определение элементов тетралогии. **Элементы тетралогии** — устройства, выполняющие функцию (операцию) тетралогии над входными сигналами

(операндами, данными) и предназначенные для обработки информации в цифровой форме.

В рамках данной работы цифровой сигнал для элементов тетралогии, как сигнал данных, у которого каждый из представляющих параметров описывается функцией дискретного времени и конечным множеством возможных значений, будет задаваться последовательностью сигналов высокого — «В» и низкого — «Н» уровней цифрового сигнала, что соответствует значениям единицы и нуля двоичной логики. Последнее обосновано тем, что элементы тетралогии, как составляющие постбинарного компьютеринга, реализуются на базе современных двоичных компьютерных систем. При этом состояния тетралогии кодируются парой двоичных разрядов [5, с. 260–273]: «00» — для неопределенности А; «01» — для тетрануля; «10» — для тетраединицы; «11» — для множественности М.

Проектирование компьютерных компонентов для реализации логических операций тетралогии включает в себя следующие этапы:

1. Формирование классической таблицы истинности для тетрафункции с учетом замены каждого тетрита парой двоичных значений.
2. Формирование аналитического выражения, являющегося основой для синтеза с использованием канонической нормальной формы представления логических функций [6].
3. Синтез комбинационной схемы и ее проектирование в среде разработки, моделирования и верификации проектов для программируемых логических интегральных схем Active-HDL компании Aldec, с использованием

языка описания аппаратуры интегральных схем VHDL [7].

В табл. 1 представлена таблица истинности для унарных тетрафункций: инверсия (pNOT), максимизация и минимизация неопределенности (MAX_A, MIN_A) и множественности (MAX_M, MIN_M) [3, с. 136], а в табл. 2 — для бинарных тетрафункций: конъюнкция (pAND), дизъюнкция (pOR) и исключающее ИЛИ (сумма по модулю 2, pXOR) [5, с. 256–257, с. 270–272].

Из табл. 1 получаем следующие аналитические выражения (все операции в данных выражениях — операции булевой алгебры):

$$q_0 = a_0 \vee \bar{a}_1 = \bar{a}_0 \wedge a_1; \quad (1)$$

$$q_1 = a_1; \quad (2)$$

$$r_0 = a_0; \quad (3)$$

$$r_1 = \bar{a}_0 \wedge a_1 = \overline{a_0 \vee \bar{a}_1}; \quad (4)$$

$$s_0 = a_0; \quad (5)$$

$$s_1 = \bar{a}_0 \vee a_1 = \overline{a_0 \wedge \bar{a}_1}; \quad (6)$$

$$v_0 = a_0 \wedge \bar{a}_1 = \overline{\bar{a}_0 \vee a_1}; \quad (7)$$

$$v_1 = a_1; \quad (8)$$

$$w_0 = \bar{a}_0; \quad (9)$$

$$w_1 = \bar{a}_1. \quad (10)$$

На основании выражений (1–10) получены комбинационные схемы, которые приведены на рис. 1. На рис. 2 приведены созданные в Active-HDL компоненты для унарных функций тетралогии с демонстрацией результатов моделирования.

Таблица 1. Таблица истинности для унарных тетрафункций (в скобках приведены значения тетритов)

| Входные значения | Выходные значения | | | | |
|------------------|-------------------|-----------------|-----------------|-----------------|---------------|
| | минимизации | | максимизации | | инверсии |
| | $q = MIN_A(a)$ | $r = MIN_M(a)$ | $s = MAX_A(a)$ | $v = MAX_M(a)$ | $w = pNOT(a)$ |
| $a[1:0]$ | $q[1:0]$ | $r[1:0]$ | $s[1:0]$ | $v[1:0]$ | $w[1:0]$ |
| 00 (A) | 01 (0) | 00 (A) | 10 (1) | 00 (A) | 11 (M) |
| 01 (0) | 01 (0) | 01 (0) | 01 (0) | 01 (0) | 10 (1) |
| 10 (1) | 10 (1) | 10 (1) | 10 (1) | 10 (1) | 01 (0) |
| 11 (M) | 11 (M) | 01 (0) | 11 (M) | 10 (1) | 00 (A) |

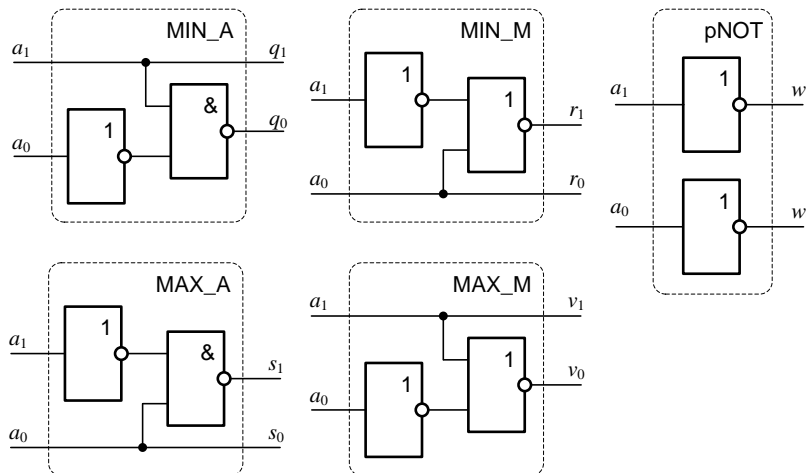


Рисунок 1 — Комбинационные схемы логических элементов, реализующих унарные операции тетралогии

Из табл. 2 получаем следующие аналитические выражения (все операции также являются операциями булевой алгебры):

$$x_0 = a_0 \vee b_0; \quad (11)$$

$$x_1 = a_1 \wedge b_1; \quad (12)$$

$$y_0 = a_0 \wedge b_0; \quad (13)$$

$$y_1 = a_1 \vee b_1; \quad (14)$$

$$z_0 = (\bar{a}_0 \wedge \bar{b}_0) \vee (a_0 \wedge b_0); \quad (15)$$

$$z_1 = (\bar{a}_1 \wedge b_1) \vee (a_1 \wedge \bar{b}_1). \quad (16)$$

На основании аналитических выражений (11–14) для бинарных тетрафункций $pAND$ и pOR спроектированы комбинационные схемы, которые приведены на рис. 3. На рис. 4 приведены созданные в Active-HDL компоненты для конъюнкции и дизъюнкции тетралогики с демонстрацией результатов моделирования.

Реализация функции $pXOR$ не нуждается в синтезе отдельного логического элемента, так как может быть реализована с использованием уже разработанных компонентов $pAND$, pOR и $pNOT$, поскольку

$$\begin{aligned} pXOR(a, b) &= \\ &= pOR(pAND(pNOT(a), b), \\ & \quad pAND(a, pNOT(b))). \end{aligned} \quad (17)$$

Справедливость выражения (17) может быть доказана подстановкой полученных ранее зависимостей (9–14) для функций $pAND$, pOR и $pNOT$. Так, для пар значений (a_0, b_0) и (a_1, b_1) получаем

$$\begin{aligned} pXOR(a_0, b_0) &= \\ &= pOR(pAND(\bar{a}_0, b_0), pAND(a_0, \bar{b}_0)) = \\ &= pOR(\bar{a}_0 \vee b_0, a_0 \vee \bar{b}_0) = \\ &= (\bar{a}_0 \vee b_0) \wedge (a_0 \vee \bar{b}_0) = \\ &= (\bar{a}_0 \wedge \bar{b}_0) \vee (a_0 \wedge b_0). \end{aligned} \quad (18)$$

$$\begin{aligned} pXOR(a_1, b_1) &= \\ &= pOR(pAND(\bar{a}_1, b_1), pAND(a_1, \bar{b}_1)) = \\ &= pOR(\bar{a}_1 \wedge b_1, a_1 \wedge \bar{b}_1) = \\ &= (\bar{a}_1 \wedge b_1) \vee (a_1 \wedge \bar{b}_1). \end{aligned} \quad (19)$$

Сопоставляя результаты выражений (18), (19) и (15), (16), очевидно, что $pXOR(a_0, b_0) = z_0$ и $pXOR(a_1, b_1) = z_1$. Учитывая, что $z = pXOR(a, b)$ (табл. 2), справедливость выражения (17) доказана.

На рис. 5 в Active-HDL выполнена практическая реализация зависимости (17) с демонстрацией результатов моделирования.

Таблица 2. Таблица истинности для бинарных тетрафункций
(в скобках приведены значения тетритов)

| Входные значения | | Выходные значения | | |
|------------------|----------|---|--|--|
| | | постбинарной конъюнкции $x = pAND(a, b)$ | постбинарной дизъюнкции $y = pOR(a, b)$ | постбинарной суммы по модулю 2 $z = pXOR(a, b)$ |
| $a[1:0]$ | $b[1:0]$ | $x[1:0]$ | $y[1:0]$ | $z[1:0]$ |
| 00 (A) | 00 (A) | 00 (A) | 00 (A) | 01 (0) |
| 00 (A) | 01 (0) | 01 (0) | 00 (A) | 00 (A) |
| 00 (A) | 10 (1) | 00 (A) | 10 (1) | 11 (M) |
| 00 (A) | 11 (M) | 01 (0) | 10 (1) | 10 (1) |
| 01 (0) | 00 (A) | 01 (0) | 00 (A) | 00 (A) |
| 01 (0) | 01 (0) | 01 (0) | 01 (0) | 01 (0) |
| 01 (0) | 10 (1) | 01 (0) | 10 (1) | 10 (1) |
| 01 (0) | 11 (M) | 01 (0) | 11 (M) | 11 (M) |
| 10 (1) | 00 (A) | 00 (A) | 10 (1) | 11 (M) |
| 10 (1) | 01 (0) | 01 (0) | 10 (1) | 10 (1) |
| 10 (1) | 10 (1) | 10 (1) | 10 (1) | 01 (0) |
| 10 (1) | 11 (M) | 11 (M) | 10 (1) | 00 (A) |
| 11 (M) | 00 (A) | 01 (0) | 10 (1) | 10 (1) |
| 11 (M) | 01 (0) | 01 (0) | 11 (M) | 11 (M) |
| 11 (M) | 10 (1) | 11 (M) | 10 (1) | 00 (A) |
| 11 (M) | 11 (M) | 11 (M) | 11 (M) | 01 (0) |

Синтез унарных тетралогических операций инверсной (shift) и сдвиговой (swap) групп

Таблица 3 представляет собой таблицу истинности для унарных тетрафункций инверсной группы [3, с. 136–137]: симметричных

($SWAP_A/M$, $SWAP_0/1$) и вероятностных ($SWAP_FL$, $SWAP_TR$).

Следующие аналитические выражения получены из табл. 3 с применением в некоторых выражениях операции отрицания булевой алгебры:

$$k_0 = a_1; \quad (18)$$

$$k_1 = a_0; \quad (19)$$

$$l_0 = \bar{a}_1; \quad (20)$$

$$l_1 = \bar{a}_0; \quad (21)$$

$$m_0 = \bar{a}_0; \quad (22)$$

$$m_1 = a_1; \quad (23)$$

$$n_0 = a_0; \quad (24)$$

$$n_1 = \bar{a}_1. \quad (25)$$

Комбинационные схемы, построенные на основании выражений (18–25) приведены на рис. 6.

Для унарных логических операций сдвиговой группы определены направления «В»

(сдвиг «назад», совпадающий с направлением оси «False» двумерного логического пространства) и «F» (сдвиг «вперед», совпадающий с направлением оси «True» двумерного логического пространства) [3]. Так, логическая операция $SHIFT_{<i>B}(a)$ определяет сдвиг тетрита a «назад» на i разрядов, а операция $SHIFT_{<i>F}(a)$ — сдвиг a «вперед» на i разрядов ($i \in \square$). В табл. 4 приведены результаты выполнения операций $SHIFT_{<i>F}(a)$ и $SHIFT_{<i>B}(a)$ для всех возможных входных значениях a и при количествах сдвигов, равных $i \bmod 4$.

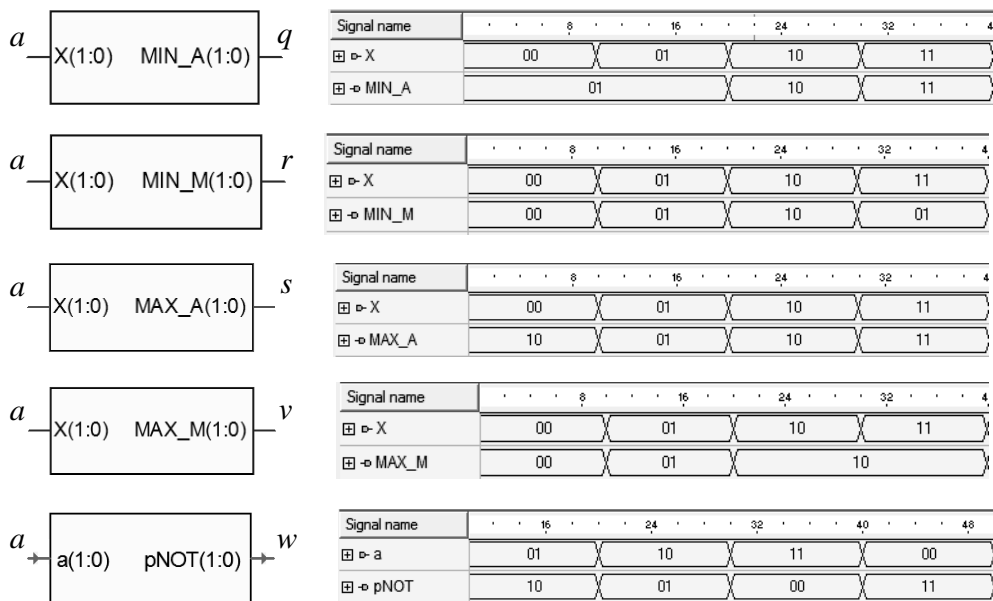


Рисунок 2 — Созданные в Active-HDL компоненты (слева), реализующие унарные функции тетралогики и диаграммы моделирования (справа)

Таблица 3. Таблица истинности для унарных тетрафункций инверсной группы (в скобках приведены значения тетритов)

| Входные значения | Выходные значения | | | |
|------------------|-----------------------|---------------------|------------------------|--------------------|
| | симметричной инверсии | | вероятностной инверсии | |
| | $k = SWAP_{A/M}(a)$ | $l = SWAP_{O/I}(a)$ | $m = SWAP_{FL}(a)$ | $n = SWAP_{TR}(a)$ |
| $a[1:0]$ | $k[1:0]$ | $l[1:0]$ | $m[1:0]$ | $n[1:0]$ |
| 00 (A) | 00 (A) | 11 (M) | 01 (O) | 10 (I) |
| 01 (O) | 10 (I) | 01 (O) | 00 (A) | 11 (M) |
| 10 (I) | 01 (O) | 10 (I) | 11 (M) | 00 (A) |
| 11 (M) | 11 (M) | 00 (A) | 10 (I) | 01 (O) |

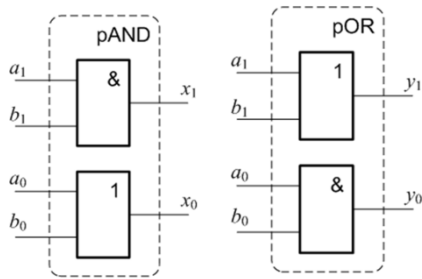


Рисунок 3 — Комбинационные схемы логических элементов, реализующих конъюнкцию и дизъюнкцию тетралогии

Сопоставляя данные таблиц 1, 3 и 4, можно убедиться в справедливости равенств (26) и (27).

$$\begin{aligned}
 &SHIFT_{<i>F(a)} = \\
 &\begin{cases} a, & \text{if } i \bmod 4 = 0, \\ SWAP_0/1(SWAP_FL(a)), & \text{if } i \bmod 4 = 1, \\ pNOT(a), & \text{if } i \bmod 4 = 2, \\ SWAP_0/1(SWAP_TR(a)), & \text{if } i \bmod 4 = 3. \end{cases} \quad (26)
 \end{aligned}$$

$$\begin{aligned}
 &SHIFT_{<i>B(a)} = \\
 &\begin{cases} a, & \text{if } i \bmod 4 = 0, \\ SWAP_0/1(SWAP_TR(a)), & \text{if } i \bmod 4 = 1, \\ pNOT(a), & \text{if } i \bmod 4 = 2, \\ SWAP_0/1(SWAP_FL(a)), & \text{if } i \bmod 4 = 3. \end{cases} \quad (27)
 \end{aligned}$$

Из (26) и (27) следует, что для аппаратной реализации операций $SHIFT_{<i>F(a)}$ и $SHIFT_{<i>B(a)}$ нужно использовать компоненты $SWAP_0/1$, $SWAP_FL$, $SWAP_TR$ и $pNOT$. Также можно сделать заключение, что тождественное равенство

$$SHIFT_{<i>F(a)} \equiv SHIFT_{<j>B(a)} \quad (28)$$

справедливо, тогда и только тогда, когда значения i и j связаны соотношением (29).

$$j = \begin{cases} 0, & \text{if } i \bmod 4 = 0, \\ 4 - i \bmod 4, & \text{if } i \bmod 4 \neq 0. \end{cases} \quad (29)$$

Рисунок 7 иллюстрирует равенства (26) и (27) для нетривиальных композиций функции при $i \bmod 4 = \{1, 3\}$ с учетом тождества (28).

Очевидно, что схемы, приведенные на рис. 7, можно упростить так, чтобы использовать только логические элементы $SWAP_FL$ и $SWAP_TR$ (рис. 8).

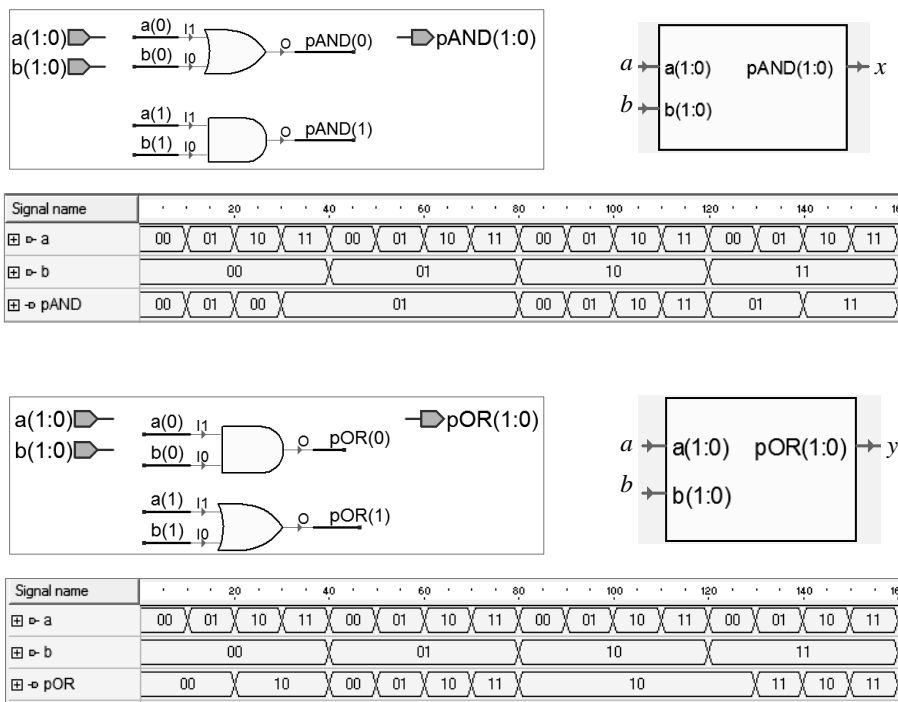


Рисунок 4 — Созданные в Active-HDL компоненты, реализующие унарные функции тетралогии: конъюнкцию (вверху) и дизъюнкцию (внизу)

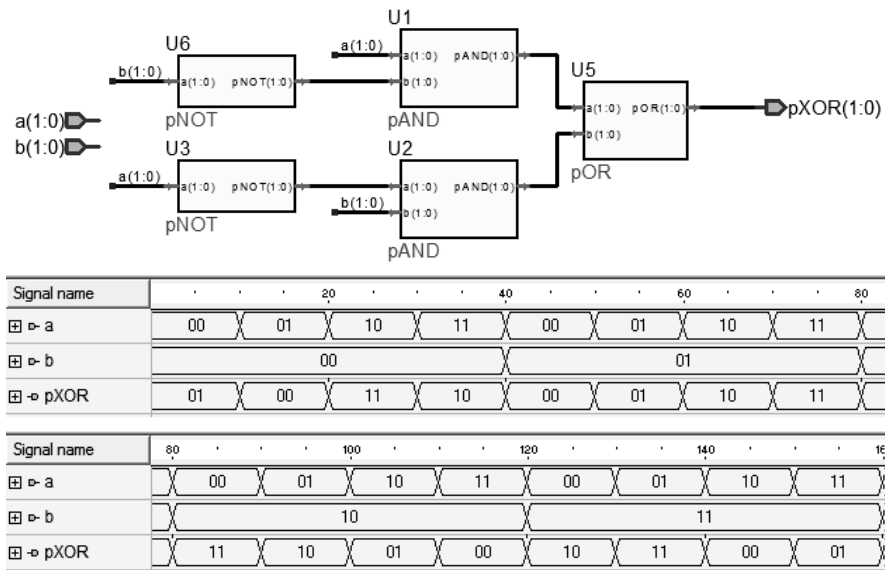


Рисунок 5 — Реализация операции тетралогии pXOR с использованием компонентов pAND, pOR и pNOT

Таблица 4. Таблица истинности для унарных тетрафункций сдвиговой группы (в скобках приведены значения тетритов)

| Входные значения | Сдвиг | Выходные значения | |
|------------------|-------------|------------------------|-------------------------|
| | | $f' = SHIFT_{<i>F(a)}$ | $f'' = SHIFT_{<i>B(a)}$ |
| $a[1:0]$ | $i \bmod 4$ | $f'[1:0]$ | $f''[1:0]$ |
| 00 (A) | 0 | 00 (A) | 00 (A) |
| 01 (O) | | 01 (O) | 01 (O) |
| 10 (I) | | 10 (I) | 10 (I) |
| 11 (M) | | 11 (M) | 11 (M) |
| 00 (A) | 1 | 01 (O) | 10 (I) |
| 01 (O) | | 11 (M) | 00 (A) |
| 10 (I) | | 00 (A) | 11 (M) |
| 11 (M) | | 10 (I) | 01 (O) |
| 00 (A) | 2 | 11 (M) | 11 (M) |
| 01 (O) | | 10 (I) | 10 (I) |
| 10 (I) | | 01 (O) | 01 (O) |
| 11 (M) | | 00 (A) | 00 (A) |
| 00 (A) | 3 | 10 (I) | 01 (O) |
| 01 (O) | | 00 (A) | 11 (M) |
| 10 (I) | | 11 (M) | 00 (A) |
| 11 (M) | | 01 (O) | 10 (I) |

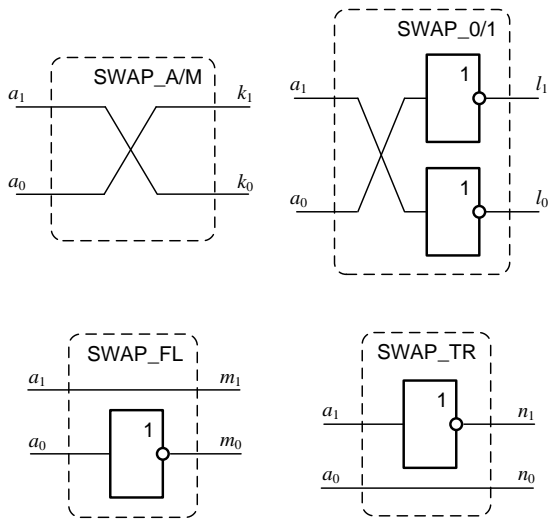


Рисунок 6 — Комбинационные схемы логических элементов, реализующих операции тетралогии инверсной группы

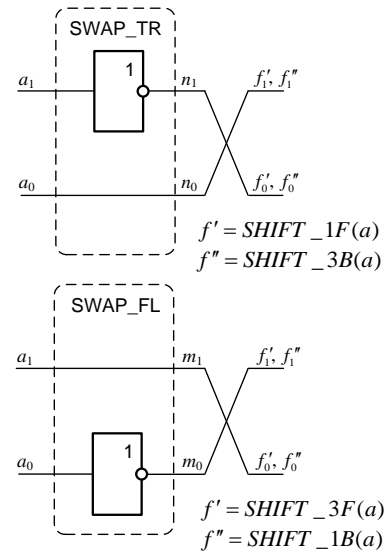


Рисунок 8 — Результат упрощения комбинационных схем, реализующих операции тетралогии сдвиговой группы

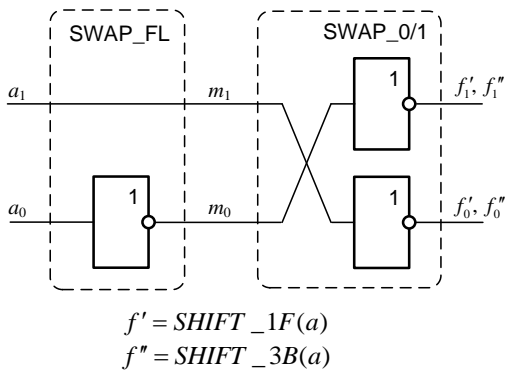


Рисунок 7 — Комбинационные схемы, реализующие операции сдвиговой группы

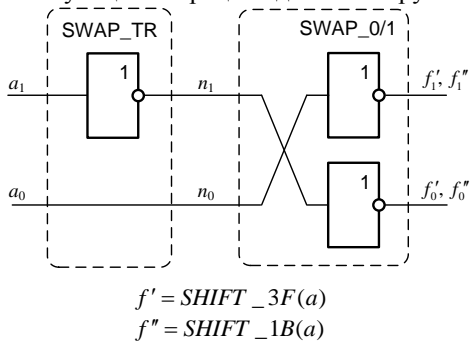


Рисунок 7 (продолжение) — Комбинационные схемы, реализующие операции сдвиговой группы

Выводы

Полученные компьютерные компоненты рассмотренных в [3] операций тетралогии могут выступать в качестве элементарного базиса для реализации специализированных компонентов, способных реализовывать базовые операции тетраарифметики, т. е. производить тетравычисления на базе современных компьютерных систем [5, 8].

Реализация тетраарифметики и тетравычислений в современных бинарных компьютерах является весомым аргументом в пользу перехода к постбинарному компьютерингу [9] и делает возможным создание вычислительных моделей, способных осуществлять достоверные вычисления и представлять в вычисляемом виде множество противоречий окружающего мира.

Литература

1. Аноприенко А. Я. Тетралогика и тетракоды / А. Я. Аноприенко // Сборник трудов факультета вычислительной техники и информатики. Вып. 1. — Донецк: ДонГТУ. — 1996. С. 32–43.
2. Аноприенко А. Я. Археомоделирование: модели и инструменты докомпьютерной эпохи / А. Я. Аноприенко. — Донецк: УНИТЕХ, 2007 — 318 с., ил.
3. Иваница С. В. Особенности реализации операций тетралогии / С. В. Иваница, А. Я. Аноприенко. // Научные труды Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2011). Выпуск

13 (185). — Донецк: ДонНТУ, 2011. С. 134–140.

4. Иваница С. В. Реализация логических операций над элементами тетракодов / С. В. Иваница, А. Я. Аноприенко // «Інформаційні управляючі системи та комп'ютерний моніторинг» (ІУС КМ – 2011) // Матеріали ІІ Всеукраїнської науково-технічної конференції студентів, аспірантів і молодих учених — 11–13 квітня 2011 г. Т.2. Донецьк, ДонНТУ. — 2011. С. 198–202.

5. Аноприенко А. Я. Тетралогіка, тетравычисления и ноокомпьютинг. / А. Я. Аноприенко, С. В. Иваница — Донецк, ДонНТУ, УНИТЕХ, 2012. — 308 с.

6. Баркалов А. А. Прикладная теория цифровых автоматов. / А. А. Баркалов, Л. А. Титаренко — Донецк: Доннту, Технопарк

ДонНТУ УНИТЕХ, 2010. — 320 с.

7. Бибило П. Н. Основы языка VHDL: Учебное пособие. / П. Н. Бибило. — Изд. 5-е. — М.: Книжный дом «ЛИБРОКОМ», 2012. — 328 с.

8. Anopriyenko A., Ivanitsa S., Hamzah A. Postbinary calculations as a machine-assisted realization of real interval calculations / International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE), 2 (4), July–August 2013, P. 91–94.

9. Аноприенко А. Я. Постбинарный компьютеринг и интервальные вычисления в контексте кодо-логической эволюции. / А. Я. Аноприенко, С. В. Иваница — Донецк, ДонНТУ, УНИТЕХ, 2011. — 248 с.

Иваница С. В. Проектирование компьютерных компонентов для реализации операций тетралогик. Выполнен синтез элементов тетралогик, выполняющих тетрафункции над входными сигналами (операндами, данными). Предложены модели компьютерных компонентов, выполненных в среде Active-HDL. Определены зависимости между унарными сдвиговыми и инверсными логическими операциями тетралогик с возможностью универсального использования синтезированных элементов.

Ключевые слова: тетракод, тетрит, тетралогіка, логические элементы, постбинарный компьютеринг.

Ivanitsa S. Designing of computer components for realization of tetralogic operations. Synthesis of tetralogical elements performing tetrafunctions over input signals is performed. Computer components models made in the Active-HDL environment are proposed. Dependencies between unary shift and inverse logical operations of tetralogy with the possibility of universal use determined synthesized elements.

Keywords: tetracode, tetrir, tetralogic, logical elements, postbinary computing.

*Статья поступила в редакцию 25.09.2017 г.
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом*

УДК 004.0

Модель поддержки принятия решений в управлении жизненным циклом мобильного приложения

Искра Е.А.

ГОУВПО «Донецкий национальный технический университет»
iskra_helen@mail.ru

Искра Е.А., Модель поддержки принятия решений в управлении жизненным циклом мобильного приложения. Статья посвящена актуальным вопросам управления жизненным циклом мобильных приложений. Определена необходимость использования комбинированных подход системно-динамического и агентного моделирования в процессе управления жизненным циклом мобильного приложения. Разработана модель поддержки принятия решений в управлении жизненным циклом мобильного приложения в среде имитационного моделирования AnyLogic 8.

Ключевые слова: *мобильное приложение, жизненный цикл, система взаимодействия, система поддержки принятия решений, системно-динамическое моделирование, агентное моделирование*

Развитие информационных технологий и массовое распространение мобильных устройств дали толчок к развитию нового рынка информационного сектора – рынка мобильных приложений. Мобильные приложения являются одним из главных приоритетов последних лет в сфере информационных технологий. Растет количество разработчиков и заказчиков, увеличивается количество приложений и пользователей, а также все большее число предприятий понимают необходимость создания конкурентоспособных приложений для повышения конкурентоспособности.

Мобильное приложение является программным продуктом, которому присущи свои особенности жизненного цикла.

Такое понятие, как жизненный цикл на сегодняшний день можно встретить в различных направлениях деятельности. И в зависимости от конкретного направления, значение может отличаться в деталях и нюансах, однако основная суть заключается в определенной последовательности этапов и процессов. Так же существует множество различных методов управлением жизненным циклом. Весомый вклад в понимания управления моделями жизненного цикла внесли Берг Д. Б., Ульянова Е. А., Добряк П. В. [1].

Итак, для управления жизненным циклом существует масса решений, моделей и стандартов которые описывают различные этапы и процессы. Однако для такого информационного продукта, как мобильные приложения, крайне мало фундаментальных исследований, а те, которые есть – быстро

теряют актуальность, ввиду динамического развития данного направления. Кроме того, мобильные приложения отличаются своим назначением, функциональностью и другими характеристиками. Поэтому более ранние модели управления подробного рода продуктами далеко не всегда применимы. Именно поэтому, мобильные приложения следует выделить в отдельную категорию и определить для них свои методы управления.

В жизненном цикле мобильного приложения, самой важной частью является фаза продвижения и эксплуатации. Для более эффективного управления жизненным циклом данной фазы, была разработана модель поддержки принятия решений, которая включает в себя комбинированных подход системно-динамического и агентного моделирования в среде имитационного моделирования AnyLogic 8 [2]. Так же был рассмотрен процесс создания надстройки к модели для моделирования мероприятий коррекции жизненного цикла в частных случаях, что позволяет переходить от модели к конкретным управленческим решениям.

Модели поддержки принятия решений по управлению жизненным циклом мобильного приложения была разработана на примере мобильного приложения «Единство», которое включает в себя три группы агентов взаимодействия, что затрудняет процесс продвижения данного продукта.

Для системы «Единство» автором была разработана имитационная модель, описывающая поведение пользователей

мобильного приложения, с учетом местных, региональных особенностей.

Рассмотрим базисную часть модели, а так же процесс моделирования частного случая, по которому требуется поддержка в принятии решения.

Имитационная модель объединяет в себе агентный и системно-динамические подходы к моделированию. В модели созданы различные популяции агентов, которые взаимодействуют между собой, это потенциальные и активные пользователи системы, а так же потенциальные и действующие партнёры системы. В модели представлены так же органы власти, имеющие

отношение к целевой аудитории системы. Все агенты из разных популяций взаимодействуют между собой и имеют свои диаграммы состояний. Системная динамика в данной модели отвечает за контроль популяций агентов, что позволяет эффективнее использовать модель в условиях повышенной неопределенности.

Рассмотрим главный экран модели на рисунке 1, на нем располагается все доступные в базисном исполнении основные элементы управления и визуальной составляющей. Органы управления для частных случаев моделирования так же будут выноситься на данный экран в последующих надстройках.

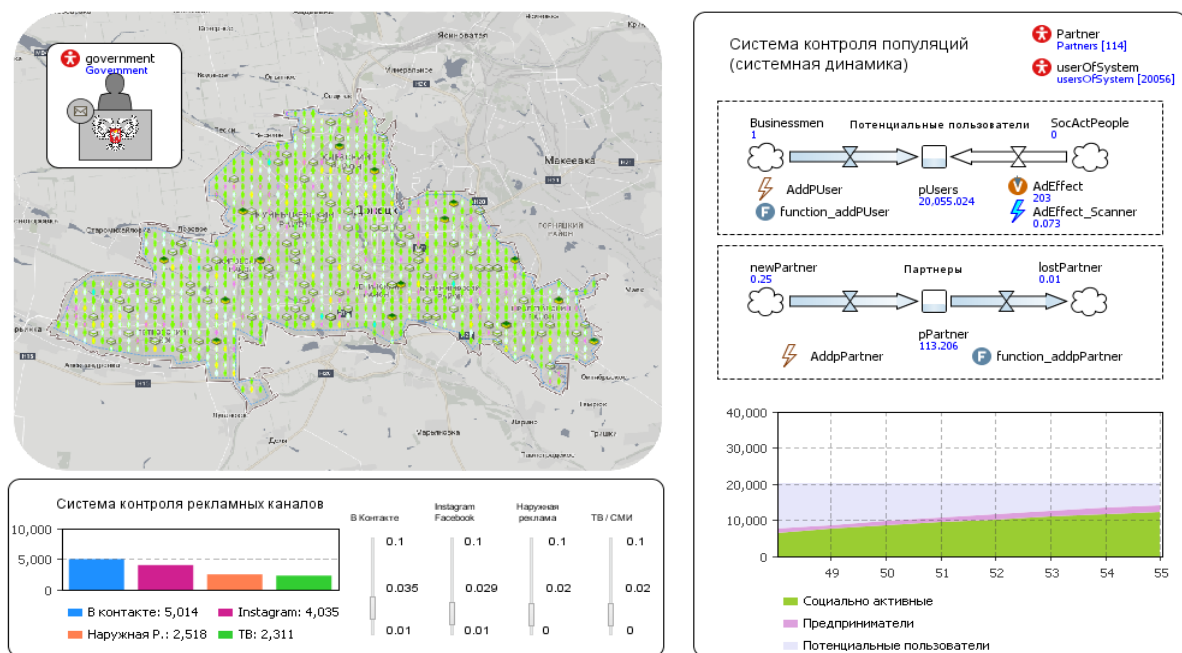


Рисунок 1 - Базисное исполнение имитационной модели системы «Единство»

На рисунке 1 представлено три блока: система контроля популяций, система контроля рекламных каналов, а так же визуализация состояний агентов в регионе.

Блок контроля популяции в первую очередь состоит из элементов системной динамики, таких как накопители (уровни) и потоки. Значение накопителей задают размер популяций агентов. Влияние на популяции и обратная связь от агентов осуществляются через функции, написанные на языке программирования JAVA.

Так как популяция агентов, моделирующая пользователей системы, самая динамичная, то визуализация количества агентов популяции и их основных состояний представлена в виде

временной диаграммы с накоплением. На диаграмме отражены три основных состояния агентов: потенциальный (серый цвет), активные предприниматели (сиреневый цвет), активные пользователи (зеленый цвет).

Блок контроля рекламных каналов состоит из элементов управления интенсивностью использования различных рекламных каналов, а так же диаграммы отражающей количество привлеченных пользователей с того или иного канала. Данные использующиеся для построения этой диаграммы учитываются в обратной связи с блоком системной динамики, как показатель эффективности рекламы.

Блок визуализации состояний агентов в регионе состоит из карты региона и периметра по

границам города, в котором расположены популяции агентов. Агент, отвечающий за органы власти, представлен в верхнем левом углу. Данная визуализация отражает различные состояния у агентов популяции, для демонстрации тенденций происходящих в системе.

Теперь рассмотрим более подробно состояния агентов популяции пользователей системы в базисном исполнении на рисунке 2, которые так же могут быть расширены для моделирования различных частных ситуаций на протяжении всего жизненного цикла системы.

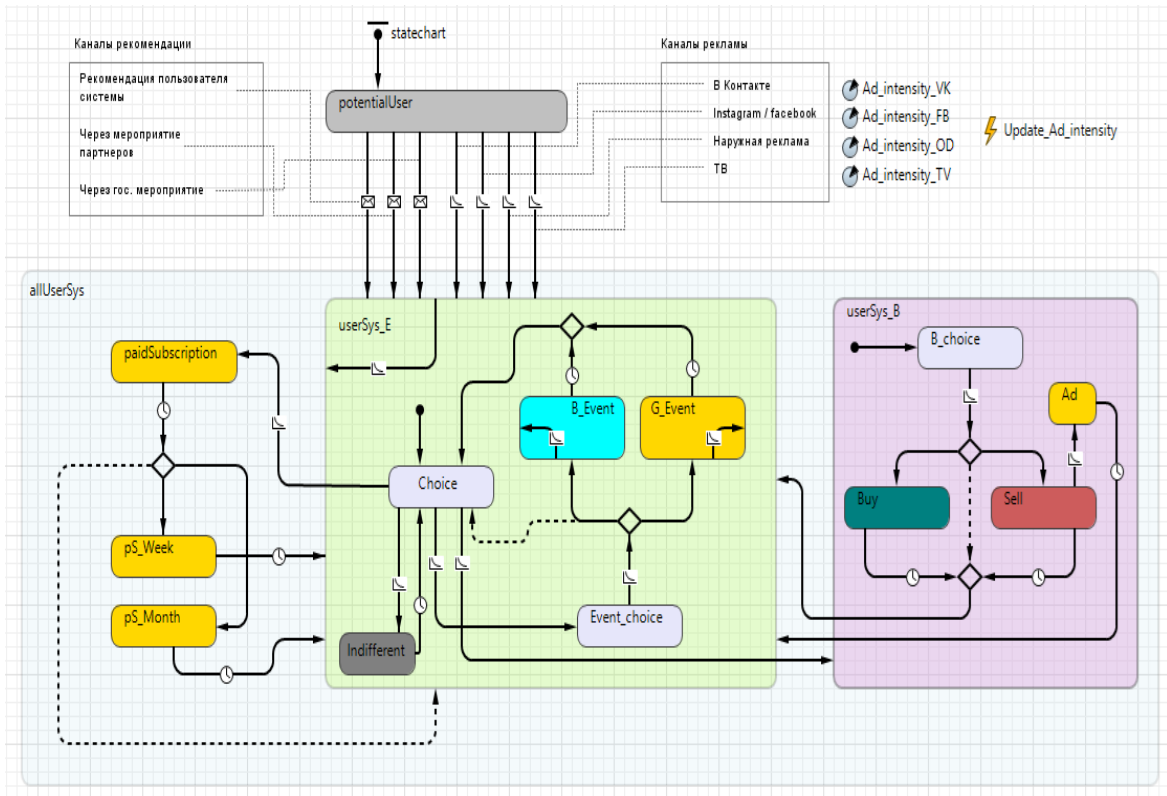


Рисунок 2 - Диаграмма состояний агента популяции пользователей

Как видно из рисунка 2, есть два ключевых состояния – потенциальный и зарегистрированный в системе пользователь. Последнее соответственно состоит из различных блоков внутренних состояний.

Переход от потенциального пользователя к зарегистрированному пользователю, обусловлен функционированием рекламных каналов, а так же различных каналов рекомендаций в процессе общения зарегистрированного пользователя со своим личным окружением.

Таким образом, у каждого агента из десятков тысяч в популяции, присутствует такая диаграмма состояний, и каждый из агентов руководствуясь логикой данной диаграммы, взаимодействует с другими таким же агентами или же агентами из другой популяции, а также

выполняет действия при переходе в то или иное состояние.

Благодаря этому можно, зная только лишь логику поведения одного агента системы, воссоздать сложные процессы, которые кажутся весьма хаотичными и сложны в описании.

Это позволяет моделировать различные ситуации и предоставить лицу принимающему решение более ясную картину последствий того или иного решения или нововведения в систему на протяжении всего жизненного цикла системы и мобильного приложения в частности.

Теперь рассмотрим моделирование частной ситуации.

Привлечение аудитории всегда будет актуальным, поэтому выберем расширение модели в данном направлении.

Предположим, что необходимо провести акцию, в рамках которой пользователи должны привести

в систему своих друзей и получить вознаграждение.

Будет существовать три рубежа, которые нужно преодолеть для получения вознаграждения, это привести три, пять и десять друзей в систему.

Для того, чтобы смоделировать этот процесс в рамках уже существующей модели, необходимо создать дополнительную надстройку к ней. Необходимо в диаграмме агента из популяции пользователей создать новый блок, с набором различных состояний описывающий участи в такой акции (рис. 3).

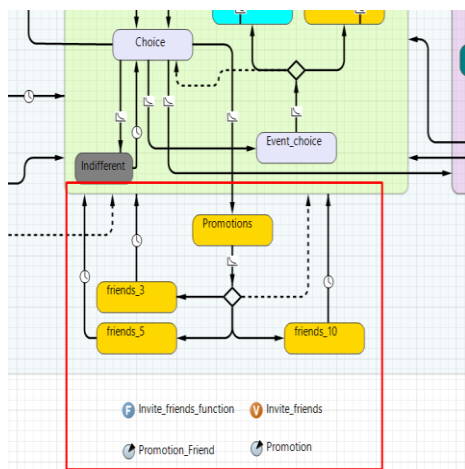


Рисунок 3 - Надстройка для диаграммы состояний агента популяции пользователей системы "Единство" (разработано автором)

Прежде чем создавать надстройку необходимо задать логику поведения агента согласно условиям акции, а затем создать интерфейс для управления данной акцией с главной диаграммы.

Для управления акцией создадим две кнопки, «Начать акцию» и «Завершить акцию», диаграмму для отображения результатов акции в процессе имитационного моделирования, а так же несколько управляющих параметров.

Результат таких действий отражен на рисунке 4.

Диаграмма, в данном блоке, отображает количество пользователей, которые достигли определенных условий в акции, и опирается на значение переменных «friends_3», «friends_5», «friends_10», которые пополняются в момент перехода, после выполнения условия акции, в созданной ранее надстройке для данной акции в диаграмме агента популяции пользователей системы.

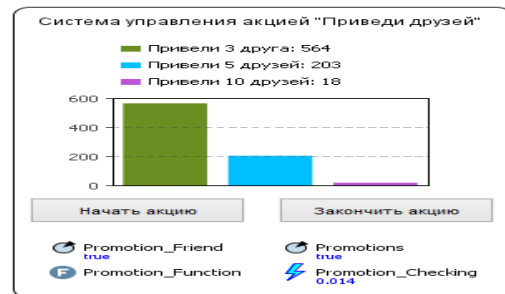


Рисунок 4 - Система управления акцией «Приведи друзей»

Общий вид главного экрана, с учетом созданной надстройки, отражен на рисунке 5.

Итак, благодаря моделированию такого частного случая, как акции «приведи друзей», лицо принимающее решение получает прогноз, какие условия, за сколько времени и в каком количестве будут выполнены на протяжении акции.

Это позволяет заранее узнать к чему готовится, какие назначать призы за достижения тех или иных условий, какие условия вводить дополнительно, а какие можно изменить.

Например, зная, что до финального условия дойдут всего несколько человек, можно сделать призы более ценными и в уже заведомо известном количестве, что позволит увеличить охват аудитории и добиться более высокой конверсии, предварительно запустив имитационную модель и скорректировав условия по тем решениям, на которые нужен прогноз.

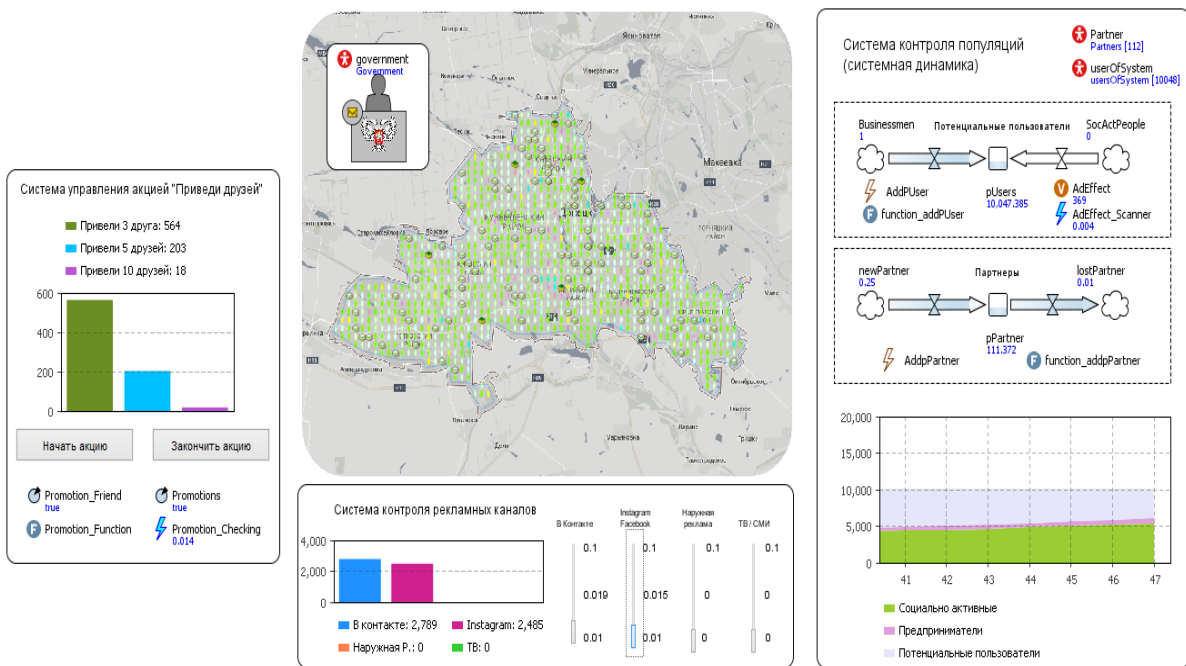


Рисунок 5 - Имитационная модель системы «Единство» в контуре системы поддержки принятия решения

Таким образом, создавая различные надстройки к модели, можно получать прогнозы для самых различных планируемых событий на различных этапах жизненного цикла. Что делает данную модель гибким и мощным инструментом в процессе поддержки принятия решений.

Использование разработанной имитационной модели поддержки принятия решений позволит планировать долгосрочные и не линейные мероприятия по управлению жизненным циклом, а универсальность среды моделирования и открытость к масштабированию имитационной модели, позволяет достраивать и планировать как глобальные изменения и влияние внешних факторов, так и мелкие корректировки внутри самого приложения. Кроме того, данная модель

предоставляется возможность существенно расширить зону контроля и повысить эффективность управления мобильными приложениями различных видов.

1. Д. Б. Берг, Е. А. Ульянова, П. В. Добряк. Модели жизненного цикла : учеб. пособие — Екатеринбург : Изд-во Урал.ун-та, 2014. — 74 с.
2. Горбунов, А. Р. Проблемы, актуальные задачи и приоритеты в создании систем поддержки принятия решений и применении имитационного моделирования в сфере управления и бизнеса [Текст] / А. Р. Горбунов, Н. Н. Лычкина // Имитационное моделирование. Теория и практика. – СПб., 2007. – Т. 1. – С. 27–36.

Искра Е.А., Модель поддержки принятия решений в управлении жизненным циклом мобильного приложения. Статья посвящена актуальным вопросам управления жизненным циклом мобильных приложений. Определена необходимость использования комбинированного системно-динамического и агентного моделирования в процессе управления жизненным циклом мобильного приложения. Разработана модель поддержки принятия решений в управлении жизненным циклом мобильного приложения в среде имитационного моделирования AnyLogic 8.

Ключевые слова: мобильное приложение, жизненный цикл, система взаимодействия, система поддержки принятия решений, системно-динамическое моделирование, агентное моделирование

Iskra O.A., Model of Decision-Making Support for a Mobile Application Life Cycle Management The paper addresses the actual issues of mobile applications life cycle management. There is determined a necessity for use of a combined approach involving system-dynamic and agent modeling in the process of mobile applications life cycle management. A model of decision-making support for the mobile application life cycle management is developed in "AnyLogic 8" software package.

Keywords: Mobile application, life cycle, decision-making, system-dynamic simulation, agent simulation.

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

Исследование графической системы с вертикальной архитектурой для поиска пересечения луча с группой полигональных объектов

Р.В. Мальчева, Т.В. Завадская, В.И. Кудояр
Донецкий национальный технический университет
raisa_malcheva@donntu.org

Мальчева Р.В., Завадская Т.В., Кудояр В.И. Исследование графической системы с вертикальной архитектурой для поиска пересечения луча с группой полигональных объектов. Выполнен анализ архитектурной организации графических систем. Предложена модификация алгоритма поиска пересечения луча с графической сценой. Выполнено проектирование системы с вертикальной архитектурой с использованием UML. Проведены исследования параметров системы поиска пересечения луча с полигональными объектами.

Ключевые слова: трассировка лучей, архитектура, параллельный поиск, проектирование, UML, исследование.

Введение

Основным недостатком метода трассировки лучей является его трудоемкость. До недавнего времени существовали только системы с программной реализацией данного метода. Однако в последнее 10-летие усилилось внимание и к аппаратной реализации трассировки, в том числе и на FPGA [1, 2]. В полной мере особенностям трассировки лучей соответствуют массивно параллельные вычислительные способности графических процессоров NVIDIA, позволяя существенно увеличить скорость визуализации в многочисленных применениях. NVIDIA поддерживает компании в расширении применения GPU-вычислений на производстве и принимает во внимание их потребности при разработке своих будущих архитектур GPU, языков и инструментов [3, 4].

Постановка задачи

Метод обратной трассировки лучей предполагает выполнение поиска пересечения луча, проведенного из глаза наблюдателя через каждый элемент окна, соответствующий пикселю экрана, с объектами сцены. Поэтому традиционно архитектура системы, основанной на этом методе, реализует принцип распараллеливания «от выходных данных» и по классификации [5] относится, в идеале, к типу «процессор на пиксель», а фактически - к «процессор на фрагмент изображения» [6, 7], хорошо реализуемый на кластерах [8].

Цель выполненных исследований – разработка системы с реализацией параллельного поиска пересечения трассируемого луча с

несколькими полигональными объектами, а также установление зависимости времени выполнения данного этапа алгоритма от числа параллельных процессов и сложности объектов. При разработке применено многоуровневое проектирование с использованием UML [9].

Реализация блока поиска пересечения луча с полигональным объектом

На вход блока CR подаются параметры луча, заданные в системе координат объекта: Xv_o , Yv_o , Zv_o - координата начала луча; Vx_o , Vy_o , Vz_o – вектор направления луча. Объект аппроксимирован плоскими выпуклыми гранями, каждая из которых описывается массивом координат вершин $\{x[n], y[n], z[n]\}$ и нормальным вектором $N=\{nx, ny, nz\}$. Для анализа используется только 1-я по списку вершина каждой грани. В результате блок формирует признак наличия пересечения (cross), и, если пересечение есть, то номер грани, в которую входит луч (gin) и значение параметра t_0 (рис.1).

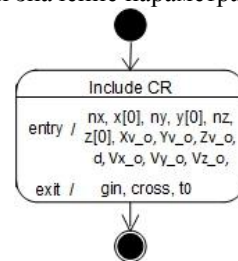


Рисунок 1 – Представление блока CR

Параметр t_0 определяет пересечение луча с многогранником и равен отношению числителя (dev) к знаменателю (div), взятому с обратным

знаком. Числитель есть подстановка точки начала луча в уравнение плоскости грани (1).

$$\text{dev} = n_x \cdot x_v^o + n_y \cdot y_v^o + n_z \cdot z_v^o + d. \quad (1)$$

При этом свободный член уравнения плоскости, d , определяется из выражения (2) подстановкой координат любой вершины грани в уравнение.

$$d = -(n_x \cdot x[0] + n_y \cdot y[0] + n_z \cdot z[0]). \quad (2)$$

Знаменатель представляет собой скалярное произведение вектора луча на нормальный вектор грани (выражение 3).

$$\text{div} = S = v_x^o \cdot n_x + v_y^o \cdot n_y + v_z^o \cdot n_z. \quad (3)$$

Все три выражения можно рассчитывать параллельно, если выражение (1) заменить на выражения (4, 5).

$$\text{dev}_0 = n_x \cdot x_v^o + n_y \cdot y_v^o + n_z \cdot z_v^o; \quad (4)$$

$$\text{dev} = \text{dev}_0 + d. \quad (5)$$

UML-диаграмма блока CR поиска параметра пересечения луча с полигональным объектом приведена на рис.2.

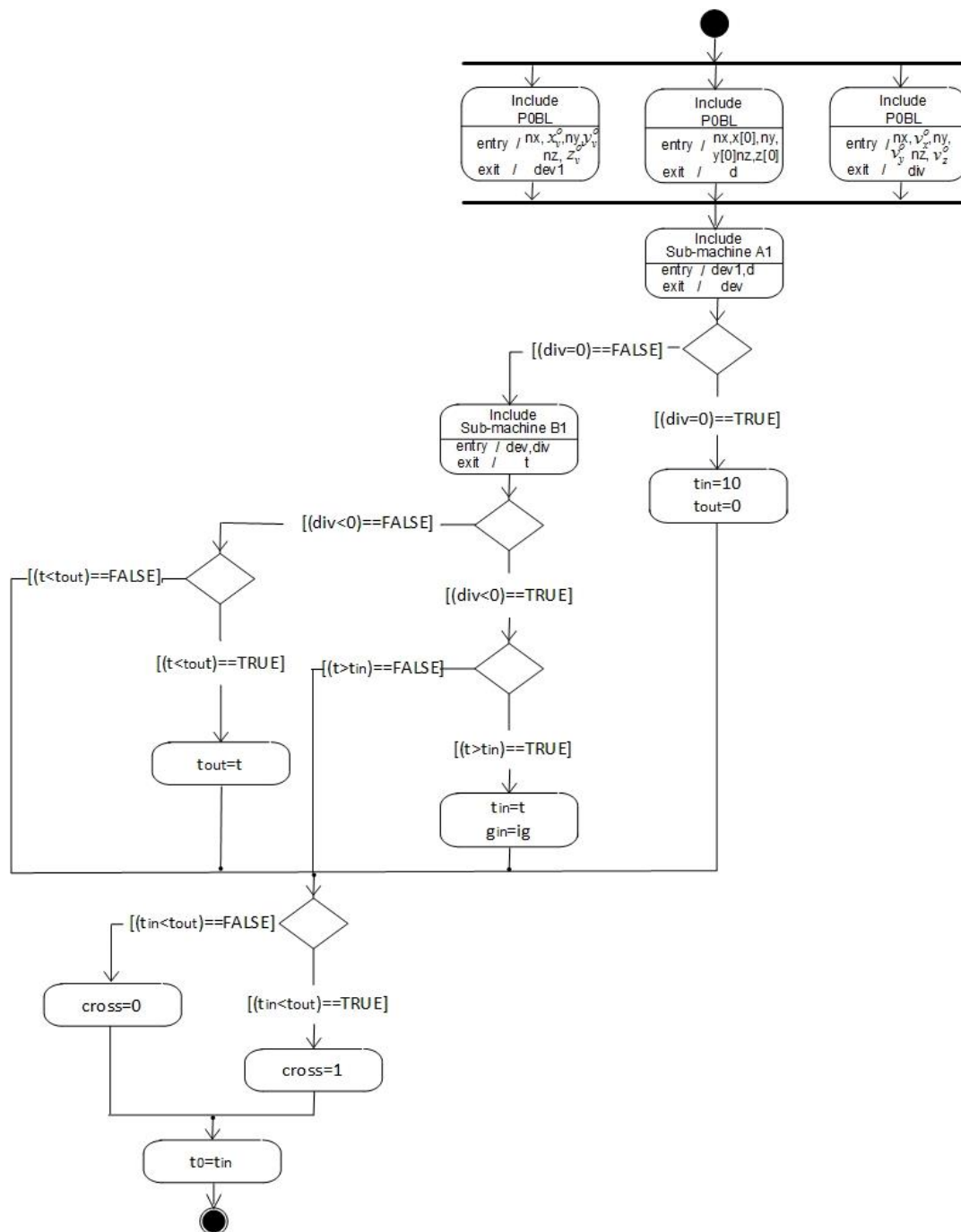


Рисунок 2 - UML-диаграмма блока CR поиска параметра пересечения луча с полигональным объектом

Очевидно, что реализация каждого выражения требует параллельного выполнения трех умножений (устройство M1) и затем двух последовательных сложений (устройство A1) [10]. Соответствующий блок назван P0BL, его диаграмма на примере вычисления скалярного произведения приведена на рис. 3.

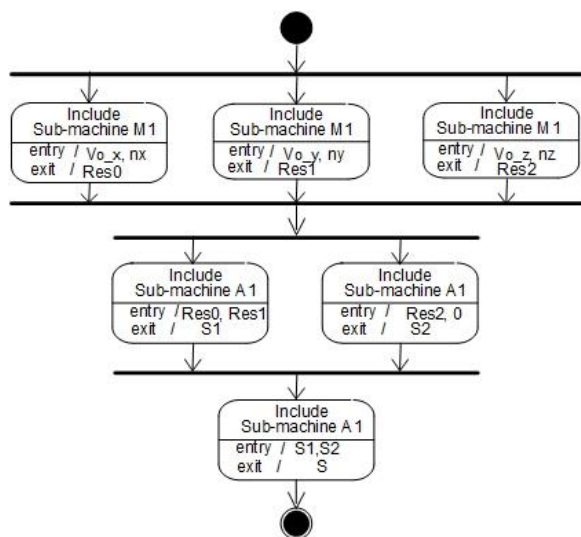


Рисунок 3 - UML-диаграмма блока вычисления скалярного произведения (P0BL)

Анализ наличия пересечения луча с объектом сводится к анализу $t_{in} < t_{out}$, т.е. расстояние до точки входа меньше, чем до точки выхода. При этом t_{in} выбирается между параметрами t видимых плоскостей ($div < 0$), а t_{out} - между параметрами невидимых.

Реализация параллельного поиска пересечения луча с группой объектов

Т.к. расчет выражений 1-5 выполняется в системе координат каждого объекта, то кроме блока CR предварительно необходимо выполнить расчет матрицы преобразования из глобальной системы и умножить на эту матрицу параметры луча. Расчет коэффициентов матрицы выполняется блоком MTR2 (рис. 4) по угловым координатам $\{\psi, \Theta, \gamma\}$ вектора P_{ok} положения каждого объекта относительно глобальной системы координат. Входными параметрами параллельных вычислительных устройств являются синусы и косинусы углов, получаемые из расположенных в постоянной памяти таблиц:

$$\begin{aligned} sp &= \sin \psi; & cp &= \cos \psi; \\ st &= \sin \Theta; & ct &= \cos \Theta; \\ sg &= \sin \gamma; & cg &= \cos \gamma. \end{aligned} \quad (6)$$

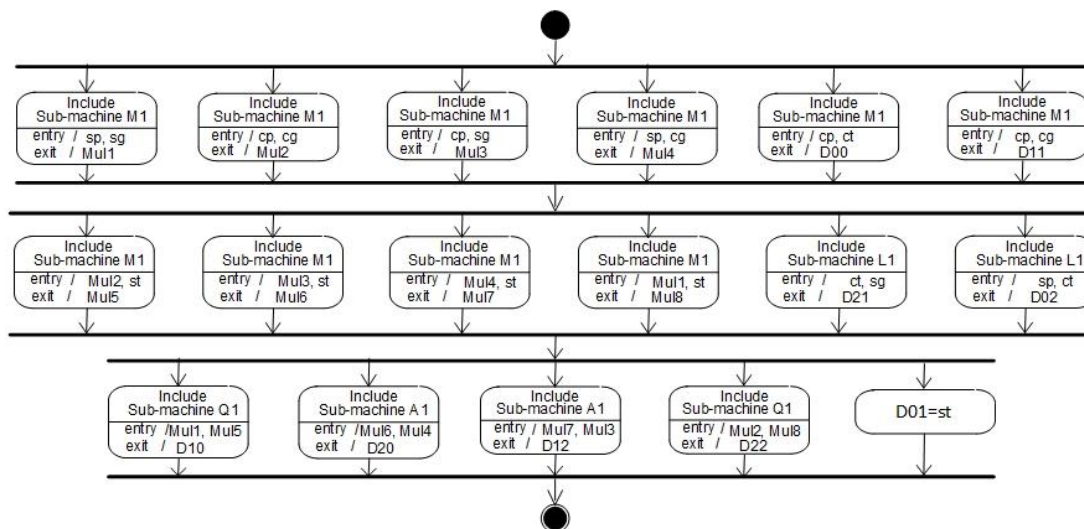


Рисунок 4 – UML-диаграмма блока MTR2 расчета матриц преобразований из глобальной системы координат в систему координат объекта

Далее блоком CC выполняется умножение координат начала луча, а блоком CCV1 – умножение параметров вектора направления луча на матрицу $D[3][3]$. Блоки CC и CCV1 включают в себя по 3 параллельных блока, аналогичных P0BL [10]. Обозначим такой объединенный блок TO.

UML-диаграмма блока CROSS приведена на рис. 5. В ней используются следующие обозначения:

hit – число пересечений луча с объектами,
gin – номер грани в точке входа луча,

i_{ob} – индекс объекта
 K_{gr} – количество групп объектов, вычисляется автоматом деления $D1$ по выражению (7):

$$K_{gr} = K_{об} / kpr, \quad (7)$$

igr – номер группы объектов, обрабатываемых параллельно,
kpr – количество параллельно анализируемых объектов,
ipr – номер объекта в группе.

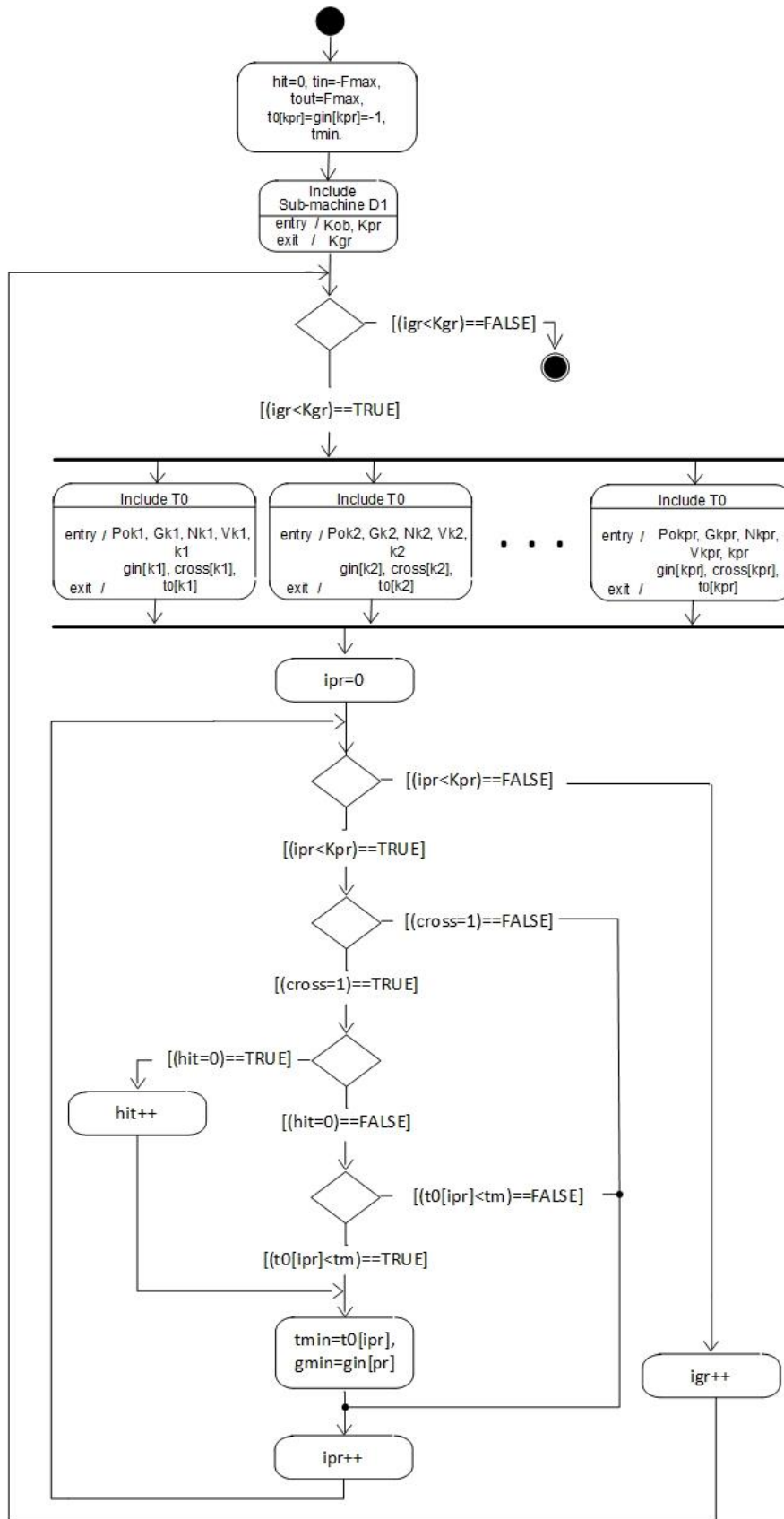


Рисунок 5 – UML-диаграмма блока CROSS поиска пересечения луча с объектами кадра

Если луч пересекает более одного объекта, то в качестве точки входа луча выбирается

минимальное значение из t_0 объектов группы. В поиске минимума участвует значение, полученное для предыдущих групп объектов.

Исследование системы

Анализ UML-диаграммы блока CROSS показывает, что система поиска пересечения луча с группой полигональных объектов имеет вертикальную архитектуру [5], в ней происходит распараллеливание «от входных данных».

Для поведения исследований на базе UML-диаграмм разработаны модели устройств с использованием языка Verilog [9, 11]. В качестве входных данных использованы базы данных по 5 миллионов полигональных объектов.

В исследованиях, результаты которых приведены на рис. 6, варьируется количество объектов в группе (соответственно, число параллельных блоков T_0 kpr, заданное по оси абсцисс). По оси ординат приведено полученное время трассировки сцены, T , машинных циклов.

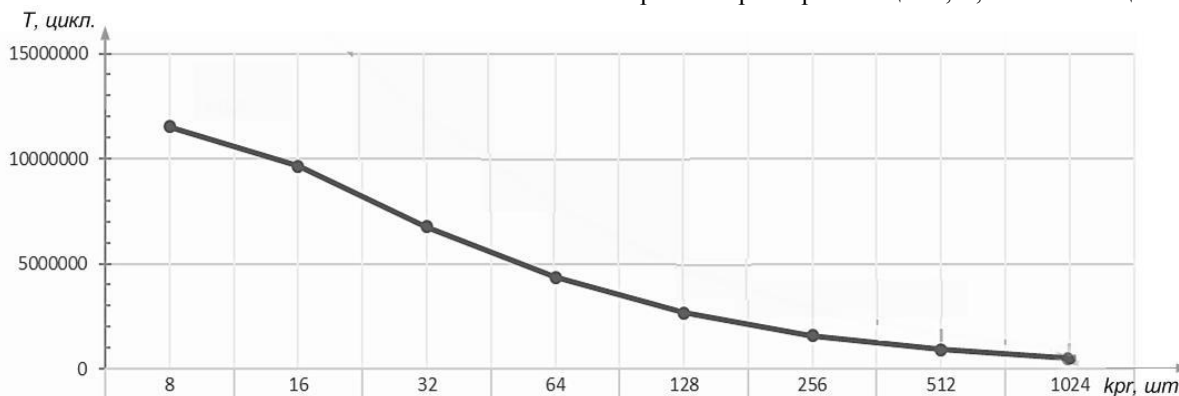


Рисунок 6 – Зависимость времени трассировки сцены от количества параллельных блоков T_0

В исследованиях, результаты которых приведены на рис. 7, время трассировки, T , задано в миллионах машинных циклов. При этом кроме количества объектов в группе варьируется

сложность полигональных объектов. Число граней принято равным 4, 8, 12, 24 и 36 – им соответствующую кривые, обозначенные $C_4 - C_{36}$.

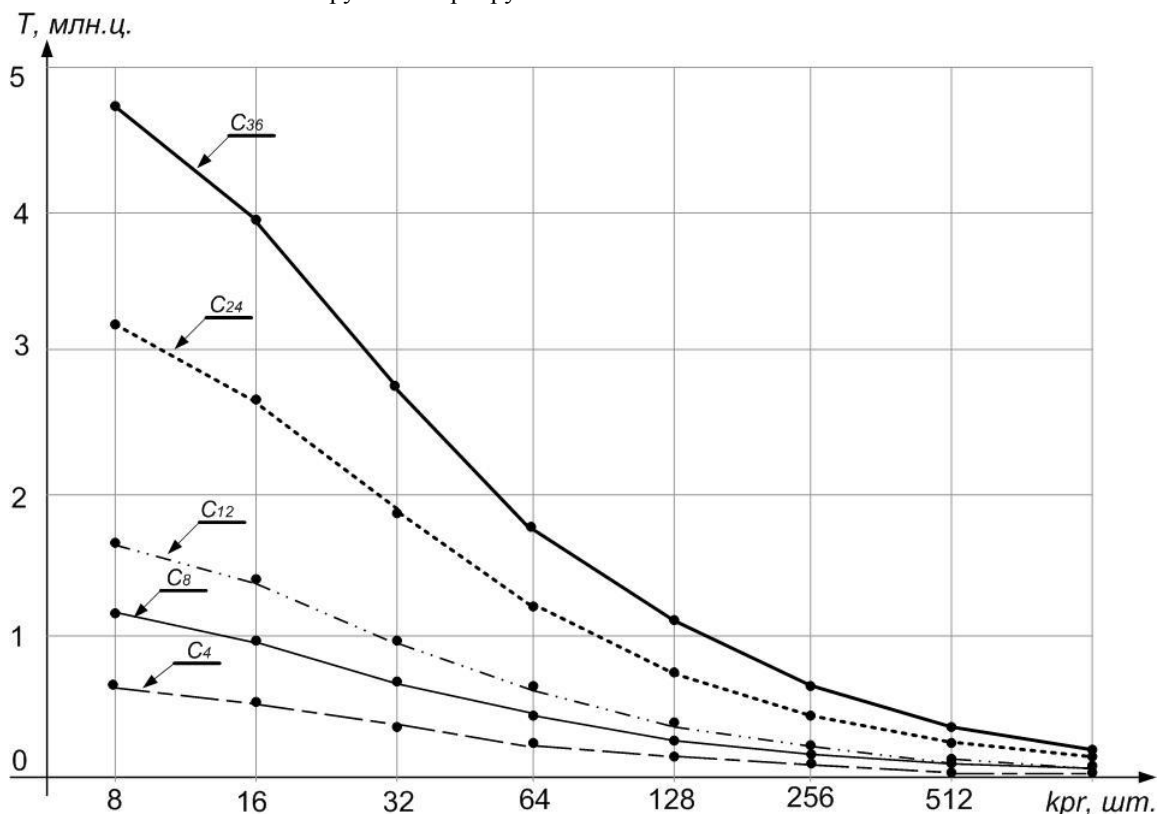


Рисунок 7 – Зависимость времени трассировки сцены от количества параллельных блоков T_0 и сложности объектов

Заклучение

Анализ зависимостей, приведенных на рис. 6 и рис. 7, показывает, что увеличение количества параллельных блоков T0 в 2 раза уменьшает время трассировки менее, чем в 2 раза. Причем, с увеличением количества объектов в группе выигрыш в производительности становится все менее заметен. Это связано с дополнительными затратами на распределение объектов по процессам вертикального конвейера. Более 512 вычислительных процессов применять не имеет смысла, что вполне соответствует размерам матриц технологии CUDA.

Список использованной литературы

1. Schmittler J. Realtime Ray Tracing of Dynamic Scenes on an FPGA Chip // Computer Science, Saarland University, 2004. - PP. 8.
2. Malcheva R. An Acceleration of FPGA-based Ray Tracer / R. Malcheva, M. Yunis // European Scientific Journal, 2014. - Vol.10, N7. - PP. 186-190.
3. Компании, использующие GPU для трассировки лучей [Электронный ресурс]– Режим доступа: <http://www.nvidia.ru/object/gpu-ray-tracing-ru.html>
4. NVIDIA обеспечивает широкую доступность физически корректного рендеринга [Электронный ресурс] – Режим доступа: <http://www.nvidia.ru/object/blog-nvidia-pbr-gtc-ru.html>
5. Computer Graphics: Principles and Practice, Third Edition / John F. Hughes, Andries van Dam, Morgan McGuire, David Sklar, James D. Foley. - Addison-Wesley, 2014. – 1209 p.
6. Bashkov E.A. Synthesis of an image of enclosing circumstances in view of a physical condition of an atmosphere / E.A. Bashkov. – Electronic simulation, 1996. –Vol.18. - N3. - PP. 45-51.
7. Bashkov E.A. Accelerating Search for Ray-Object Intersection Point in the Ray Tracing Method. - Engineering Simulation, 1996. - Vol.13. - N3. - PP. 369 - 380.
8. Мальчева Р.В. Реализация модифицированного алгоритма трассировки лучей на кластере NeClus / Р.В. Мальчева, М. Юнис // Наукові праці Донецького національного технічного університету. Серія: «Обчислювальна техніка та автоматизація». – Донецьк: ДонНТУ, 2013. – №1(24) 2013. - С. 263-268.
9. Malcheva R. Application of multilevel design on the base of UML for digital system developing // in book “Design of Digital Systems and Devices. Series: Lecture Notes in Electrical Engineering. – Springer-Verlag Berlin Heidelberg, 2011. – Vol. 79. - PP. 93-117.
10. Попова А.Н. Разработка устройства поиска пересечения луча с полигональным объектом / А.Н. Попова, А.В. Селинова, Р.В. Мальчева // «Компьютерная и программная инженерия. Сборник материалов международной научно-практической конференции студентов, аспирантов и молодых ученых» (г. Донецк, ДонНТУ, 2015 г.) – С. 242-247.
11. Хаханов В.И. Проектирование и верификация цифровых систем на кристаллах. Verilog & System Verilog / В.И. Хаханов, И.В. Хаханова, Е.И. Литвинова, О.А. Гузь. – Харьков: ХНУРЭ, 2010. – 528 с.

Malcheva R.V., Zavadskaya T.V., Kudoyar V.I. An investigation of a graphical system with a vertical architecture for searching for intersection of a ray with a group of polygonal objects. The analysis of the architectural organization of graphic systems is carried out. A modification of the search algorithm for the intersection of a ray with a graphic scene is proposed. A system with a vertical architecture is designed using UML. The parameters of a system for searching for intersection of a ray with a group of polygonal objects are investigated.

Keywords: ray tracing, architecture, parallel search, design, UML, investigation.

Мальчева Р.В., Завадська Т.В., Кудояр В.І. Дослідження графічної системи з вертикальною архітектурою для пошуку перетину промінню з групою полігональних об'єктів. Виконаний аналіз архітектурної організації графічних систем. Запропонована модифікація алгоритму пошуку перетину промінню з графічною сценою. Виконано проектування системи з вертикальною архітектурою з використанням UML. Проведені дослідження параметрів системи пошуку перетину промінню з полігональними об'єктами.

Ключові слова: трасування промінів, архітектура, паралельний пошук, проектування, UML, дослідження.

Статья поступила в редакцию 20.09.2017

Рекомендована к публикации д-ром техн. наук В.Н. Павльшином

Анализ методов распознавания эмоционального состояния человека для выявления аномального поведения в видеоаналитических системах

Т. В. Мартыненко, А. А. Щербаков, Б. В. Боднар
Донецкий национальный технический университет
kabka@list.ru

Мартыненко Т.В., Щербаков А.А., Боднар Б.В., Анализ методов распознавания эмоционального состояния человека для выявления аномального поведения в видеоаналитических системах. В данной работе рассматривается методика обнаружения аномального состояния человека путем анализа его эмоционального состояния и уровня напряженности. Приведен обзор и классификация методов автоматического распознавания эмоций. На основании проведенных исследований представлен алгоритм обнаружения аномального поведения субъекта. Для решения задачи был выбран комплексный подход, реализация которого предоставила наиболее точные результаты.

Ключевые слова: эмоции, напряженность, система видеоаналитики, аномальное поведение, определение ключевых точек, мимический признак, распознавание эмоционального состояния.

Введение

В современном мире автоматизированные системы анализа видеопотока, включающие отслеживание поведения людей, применяются в различных областях: обеспечение безопасности анализируя траектории перемещения людей [1], вспомогательная помощь при проведении оперативно-розыскных мероприятий [2], психологическое и медицинское диагностирование [3].

Одной из основных задач автоматизированного видеоанализа в данных системах является классификация поведения людей на «нормальное» и «аномальное».

Как правило «аномальное поведение» субъекта понимается как алгоритм действий, который не соответствует «нормальной» модели поведения людей в рассматриваемой ситуации, или же которая не представлена достаточным количеством примеров в обученной базе данных [4].

С позиции общей психологии и физиологии эмоциональное состояние и уровень напряженности субъекта можно рассматривать - как показатели психического, физиологического и функционального состояния человека [14].

С точки зрения невербальной психологии - как состояния, которые отражает мимика лица и прочие аспекты невербального поведения человека.

Существует достаточно много вариантов классификации эмоциональных состояний и критериев дискретизации эмоций между собой.

Несмотря на это, ни одна из классификаций не может полноценно интерпретировать все состояния случайно выбранного человека, и в этом смысле задача автоматического определения эмоций является «плохо» определенной.

Преимущество классификации Пола Экмана заключается в высокой корреляции результатов среди исследуемого набора людей по параметрам пол, национальность и возраст, и в рамках таких критериев данная классификация определяется как «универсальная». Классификация представляет собой шесть эмоций, такие как счастье, печаль, гнев, страх, удивление и отвращение (англ. happiness, sadness, anger, fear, surprise, disgust)

В итоге, получив оценку уровня напряженности и эмоционального состояния человека в определенный момент времени, а также определив модель типичного поведения и решающие правила, можно сделать вывод, является ли поведение человека аномальным.

Таким образом, целью работы является разработка алгоритма обнаружения аномального поведения человека, а также анализ методов распознавания эмоций и определение степени напряженности человека.

Разработка алгоритма обнаружения аномального поведения субъекта.

На основании проведенного анализа различных источников описывающих

распознавание объектов в видеопотоке [5, 6], а также выявление атипичного поведения субъекта [4,7- 9] и работ по способам контекстного описания видео [10,11] был составлен алгоритм обнаружения аномального поведения субъекта:

1. Первичная обработка видеоряда, извлечение отдельных кадров.
2. Обработка полученных изображений - сегментация, определение контуров объектов сцены, детектирование людей.
3. Определение шаблонов двигательной (невербальной) активности как объектов видео анализа.
4. Слежение за ранее определенными шаблонами, вычисление динамики характеристик.
5. Поиск и выделение семантических, вероятностных (статистических) и прочих характеристик двигательной активности, которые соответствуют модели поведения.
6. Принятие решения об аномальности поведения, как о совокупности количественных и семантических характеристик.
7. Вычисление ошибок, тестирование и адаптация моделей.

Модель поведения исследуемых объектов может быть:

- статистической или вероятностной моделью, к примеру:
 - динамической сетью Байеса [4] (от англ. Dinamic Bayesian network);
 - стохастической контекстно-свободной грамматикой [5] (англ. SCFG);
 - скрытой Марковской моделью [5] (от англ. Hidden Markov Model);
 - распределением плотности вероятности [7] (от англ. Probability density function);
- модель, использующая базу данных (далее БД) нормальной и аномальной активности, представленной набором изображений и их декомпозициями [8];
- моделью представления знаний [12];
- другие модели (расширения и комбинации описанных выше моделей).

Исходя из описанного в источнике [5] принятие решения может осуществляться с помощью экспертных продукционных правил, а так же путем вычисления расстояний между векторами [13] (Кульбака - Лейблера Евклида, Махаланобиса).

Описанный выше алгоритм отражает трудоемкость решения проблемы

автоматического обнаружения аномального поведения на основе полученного видеоряда. Однако для частных случаев задачи, таких как распознавание аномальной траектории движения объектов и аномальных событий [4,5], выявления аномального движения на основе информации, заложенной в видео компрессии [7], распознавание аномальных жестов и походки [8] уже существуют некоторые методы решения (реализации).

Задача распознавания аномального «эмоционального состояния» и «уровня напряженности» является актуальной и требует нахождения комплексного решения.

Анализ методов распознавания эмоций

Одним из направлений исследования в сфере распознавания образов является анализ выражения лица. В основном сложности возникают на этапе получения точных признаков и выражений лица. Все дело в том, что разница в изменении положения характерных точек между различными выражениями лица может быть незначительной, к примеру: открытый рот не обязательно будет обозначать удивление, это может быть плач или смех. Проанализированные далее методы, как правило, развились из методов распознавания лица, но могут применяться и для распознавания эмоций.

Во-первых, существующие методы распознавания используют три основных признака для распознавания - это уровень серого, частоты и движение. Различные эмоции имеют разные уровни серого на цифровом изображении лица пользователя, из-за изменения рельефа лица. Однако для нормализации входного изображения перед началом анализа необходимо осуществлять предварительную обработку изображения. Характерный признак: движение, использует информацию об изменении положения определенных точек на лице. Характерный признак: частоты, использует разницу между различными цифровыми изображениями лица в частотной области.

С другой стороны, методы распознавания можно разделить на три области [17-24] (таблица 1): холистические (целостные) и локальные распознавания, экстрагирования деформации и движения (методы, вычисляющие динамику объектов), геометрические характеристики и характеристика внешности (методы вычисляющие форму объектов).

Так методы целостных распознаваний анализируют эмоции человеческого лица в целом, чтобы затем можно было найти разницу

между разными изображениями: Principal Component Analysis (PCA), Independent Component Analysis (ICA), Fisher's Linear Discriminants (FLD), Local Feature Analysis (LFA), Fisher Actions, Hidden Markov Models, и кластерный анализ.

В свою очередь методы локальных распознаваний анализируют отдельные части лица: глаза, брови и рот. Типичные методы: Facial Actions Code System (FACS), Local PCA, Вейвлеты и Нейронные сети.

В основе методов экстрагирования деформации и движений лежит изменения формы органов на лице, когда меняются различные выражения: Active Shape Model (ASM), Point Distribution Model (PDM).

Методы, основанные на геометрических характеристиках, полагаются на форму и положение различных частей лица человека, чтобы извлечь характерные векторы, которые представляют геометрические характеристики лица.

Таблица 1 - классификация методов распознавания эмоций

| | Холистические методы | Локальные методы |
|------------------------------|--|--|
| Методы, вычисляющие форму | Классификаторы: ANN [17], SVM [21,22], Random forrest [20], LVQ [20], Adaboos1 [19], Вейвлеты и фильтры Габора [20] Преобразование Хафа [20] 2D Модели лица: AAM [20], ASM [19], EBGM [20] | Классификаторы: ANN [18], SVM [20], GA [18], Классификатор Байеса [20], Adaboost [20] Геометрические модели лица [17] Собственные вектора: PCA [20], LDA [14] Локальные гистограммы: HoG [13], LBP [19,24] |
| Методы, вычисляющие динамику | Оптический поток [20] Динамические модели [17] | 3D динамические модели [17] Статистические модели: HMM [20], DBN [20] |

Описание аббревиатур, используемых в таблице 1 представлено ниже:

- ASM - модель формы,
- EBGM - эластичный граф,
- ANN - искусственные нейронные сети,
- SVM - методы опорных векторов,
- LDA - линейный дискриминантный анализ,
- GA - генетические алгоритмы,

LVQ - векторное квантование (сеть Кохонена),

AAM - модель представления,

HoG - гистограмма направленных градиентов,

PCA - метод главных компонент,

LBP - локальные бинарные шаблоны.

Сейчас для автоматического распознавания эмоций в основном применяют комплексные подходы, общий принцип которых приведен на рисунке 1 и включает в себя следующие общие:

1. Получение изображения путем детектирования его на кадре и его адаптация (нормализация).

2. Поиск ключевых точек и особенностей лица, текстуры и формы, вычисление интенсивности проявления характеристик.

3. Опциональная обработка характеристик.

4. Непосредственно процесс классификации эмоциональных и других состояний.

5. Интерпретация и оценка результатов.

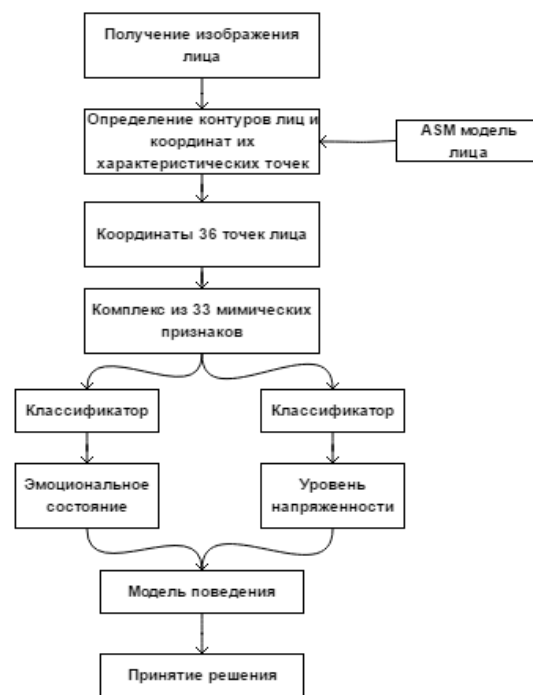


Рисунок 1 – Алгоритм обнаружения аномального поведения с помощью локального метода распознавания эмоционального состояния и уровня напряженности, вычисляющего форму.

В таблице 2 представлены краткие сведения о некоторых подходах, реализующих данный принцип и показавших наиболее точные результаты.

Таблица 2 – комплексные подходы,
 показавшие наилучшие результаты

| Извлечение особенностей | Классификатор | Точность | Эмоции | Авторы |
|---------------------------|---------------|----------|---|-------------------------|
| ASM + Мимические признаки | ANN | 77-93% | Экман + нейтральная эмоция + уровни напряженности | Кашапович и др. [17] |
| PCA + Фильтр Габора | SVM | 71-90% | Экман | N. Agrawal et al. [20] |
| Оптический поток | SVM | 74-90% | Экман | K. Anderson et al. [20] |
| Преобразование Хафа | Random Forest | 62-100% | Экман | G. Fanelli et al. [20] |

Для проведения дальнейших исследований предлагается использование метода «ASM + Мимические признаки + ANN». В этом методе на третьем шаге производится вычисление мимических признаков на основе ключевых точек лица. Данный метод позволяет классифицировать не только эмоции, но и также уровни напряженности.

Недостатки метода «ASM + Мимические признаки + ANN»:

– отсутствие вычисления динамики мимических картин;

– длительное обучение классификатора системы.

Достоинства метода «ASM + Мимические признаки + ANN»:

– использование системы классификации выражений лица (Facial Action Coding System (FACS)) П. Экмана.

– использование инвариантного к масштабу изображения лица мимического комплекса в составе 33 геометрических признаков модели лица (углов и расстояний);

– относительная простота разработки и конкурентоспособные результаты для выбранной тестируемой базы.

– устойчивость к нерегулярностям изображения лица, таким как ориентация головы и освещенность, из-за использования модели формы ASM

Заключение

В статье представлен краткий обзор современных методов автоматизированного распознавания эмоционального состояния человека, выбран наиболее подходящий комбинированный метод, выделены общие принципы построения системы и предложен комплексный алгоритм решения задачи.

Предложенное решение основано на распознавании аномального поведения человека по его текущему эмоциональному состоянию и уровню напряженности.

В последующих работах планируется описание реализации, изложение и анализ результатов работы системы основанной на методе вычисления мимических признаков и их классификации с помощью перцептронной нейронной сети и наличием блока принятия решений.

Список литературы

1. Хрулев А.А. Интегрированные автоматизированные системы безопасности для массовых мероприятий на основе биометрических технологий // Евразийский форум "Инфофорум-Евразия", 2010.

2. Спиридонов И.Н. Биометрические технологии идентификации личности и безопасность государства // Евразийский форум "Инфофорум-Евразия", 2010.

3. Спиридонов И.Н., Кашапова Л.Х. Определение эмоционального состояния по видеоизображению лица // Евразийский форум "Инфофорум-Евразия", 2012.

4. Tao Xiang, Shaogang Gong. Video Behavior Profiling for Anomaly Detection // Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.30, no.5, pp.893-908, 2008.

5. Ivanov Y.A., Bobick A.F. Recognition of multi-agent interaction in video surveillance // Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on , vol.1, no., pp.169-176 vol.1, 1999.

6. Jabri S., Duric Z., Wechsler H., Rosenfeld A. Detection and location of people in video images using adaptive fusion of color and edge information // Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.4, pp.627-630 vol.4, 2000.

7. Kiryati N., Raviv T.R., Ivanchenko Y., Rochel S. Real-time abnormal motion detection in surveillance video // Pattern Recognition, 2008. ICPR 2008. 19th International Conference on , pp.1-4, 2008.

8. Boiman O., Irani M. Detecting irregularities in images and in video // Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on , vol.1, pp.462-469, 2005.

9. Птицын Н.В. Встроенная видеоаналитика для детектирования и сопровождения объектов при помощи многомасштабных признаков. Режим доступа: <http://www.graphicon.ru/proceedings/2010/confere nce/RU/Se3/31.pdf> (дата обращения 18.02.2013)
10. Sikora T. The MPEG-7 visual standard for content description-an overview // Circuits and Systems for Video Technology, IEEE Transactions on , vol.11, no.6, pp.696-702, 2001.
11. Nevatia R., Hobbs J., Bolles B. An Ontology for Video Event Representation // Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04. Conference on, pp.119- 119, 2004
12. Abdoli F., Kahani M. Ontology-based distributed intrusion detection system // Computer Conference, 2009. CSICC 2009. 14th International CSI , pp.65-70, 2009. Fasel B., Luetin J. Automatic facial expression analysis: a survey// Pattern Recognition. 2003. vol.36, iss.1, pp.259-275, 2003.
11. Леонова А. Б., Медведев В. И. Функциональные состояния человека в трудовой деятельности. М.: [б.н.], 1981.
13. Климов С. М. Методы и модели противодействия компьютерным. Люберцы: КАТАЛИТ, 2008. 306 с.
14. Sandbach G., Zafeiriou S., Pantic M., Yin L. Static and dynamic 3D facial expression recognition: Image and Vision Computing, vol.30, iss.10, pp.683-697, 2012.
15. Кашапова Л.Х., Хрулев А.А., Спиридонов И.Н. Проектирование автоматизированной системы распознавания эмоционального состояния по мимике // Фотометрия и ее метрологическое обеспечение: Тез. докл. XVII РНТК. - М., 2008. - с. 262-264.
16. Karthigayan M., Rizon M., Nagarajan R., Yaacob S. Genetic Algorithm and Neural Network for Face Emotion Recognition // Affective Computing, 2008
17. Zhao Xingli, Zhang Hong, Xu Zhan. Expression recognition by extracting facial features of shapes and textures//Journal of Computational Information Systems. 8(8): 3377-3384, 2012
18. Azcarate A. , Hageloh F., Koen Van De S, Valenti R. Automatic facial emotion recognition // tech. rep., Universiteit Van Amsterdam, 2005.
19. Anderson K., McOwan Peter W. A real-time automated system for the recognition of human facial expressions // Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on , vol.36, no.1, pp.96-105, 2006.
20. Takiguchi T., Adachi J., Aiki Y. Video editing based on situation awareness from voice information and face emotion // In F. De Rango Ed., Digital video. Rjeka, Croatia:InTech, 2010.

Мартыненко Т.В., Щербаков А.А., Боднар Б.В., Анализ методов распознавания эмоционального состояния человека для выявления аномального поведения в видеоаналитических системах. В данной работе рассматривается методика обнаружения аномального состояния человека путем анализа его эмоционального состояния и уровня напряженности. Приведен обзор и классификация методов автоматического распознавания эмоций. На основании проведенных исследований представлен алгоритм обнаружения аномального поведения субъекта. Для решения задачи был выбран комплексный подход, реализация которого предоставила наиболее точные результаты.

Ключевые слова: эмоции, напряженность, система видеоаналитики, аномальное поведение, определение ключевых точек, мимический признак, распознавание эмоционального состояния.

Martynenko T.V., Shcherbakov A.A., Bodnar B.V. Analysis recognizing methods of the emotional state of a person to detect abnormal behavior in video analytics systems. This article deals with the technique of detecting an abnormal state of a person by analyzing its emotional state and level of tension. The review and classification of methods of automatic recognition of emotions is given. On the basis of the conducted researches the algorithm of detection of abnormal behavior of the subject is presented. to solve the problem in the future, a comprehensive approach was chosen, the implementation of which provided the most accurate results.

Keywords: emotions, tension, the system of video analytics, abnormal behavior, the definition of key points, the mimic sign, the recognition of the emotional state.

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

Исследование проблемы выбора минимального набора услуг для расширения деятельности предприятия на примере интернет-провайдера

И.В. Матях, Е.О. Савкова, О.В. Ченгарь
Донецкий национальный технический университет
iramatyakh@gmail.com

Матях И.В., Савкова Е.О., Ченгарь О.В. Исследование проблемы выбора минимального набора услуг для расширения деятельности предприятия на примере интернет-провайдера. В статье рассматривается формализация задачи принятия решения о целесообразности открытия или расширения сферы деятельности фирмы, на примере интернет-провайдера. Для этого исследованы внешние и внутренние факторы, влияющие на прибыль предприятия. Рассмотрены две модели представления задачи: в виде матрицы полезностей и матричной игры. Выделены достоинства и недостатки данных моделей, описаны методы решения задачи.

Ключевые слова: интернет-провайдер, прибыль, случайная ситуация, набор услуг, матрица полезностей, матричная игра, альтернативное решение, правило комбинаторики.

Введение

Каждый из нас, хоть раз в жизни, наблюдал за торжественным открытием различных магазинов, фирм, коммерческих организаций. Но, к сожалению, некоторые из них, вскоре прекращали свою деятельность. И вот, на месте только что открывшегося магазина остались лишь пустые помещения, которые, в настоящее время, часто можно увидеть на улицах города.

На успешность работы любой фирмы влияет огромное количество факторов, как внешних (например, регион, где расположена фирма, действие или бездействие конкурентов в данном регионе), так и внутренних (правильно ли подобраны спектр услуг, цены на предоставляемые услуги). Успешная деятельность нового или расширения сферы деятельности существующего предприятия во многом зависит от определения таких факторов и оценки их влияния на прибыль. Проблема оценки этих показателей остается актуальной на сегодняшний день. Многолетний опыт отечественных и зарубежных фирм показывает, что недооценка планирования и прогнозирования предпринимательской деятельности или же их некомпетентное выполнение приводят к экономическим затратам (убытку) или же к банкротству предприятия.

Следовательно, руководитель предприятия (менеджер) должен проанализировать множество случайных ситуаций, которые могут повлиять на работу предприятия и спрогнозировать получаемую прибыль при таких ситуациях. Только после

этого можно принять решение о целесообразности открытия или расширения фирмы.

Постановка задачи

В практической деятельности часто встречаются задачи, решение которых заключается в поиске лучшего варианта (альтернативы) по одному или нескольким критериям. В данном случае, альтернативой является выбор руководителем предприятия одного решения из множества возможных. Критерий – один из элементов задачи принятия решений, в соответствии с которым лицо, принимающее решение выбирает ту или иную альтернативу [1].

Для выбора лучшей альтернативы, необходимо выполнить анализ случайных показателей, влияющих на работу фирмы и провести оценку их влияния на доход предприятия.

Задачу принятия решения об открытии фирмы или расширения ее сферы деятельности для получения максимальной прибыли рассмотрим на примере организации, предоставляющей интернет-услуги населению.

В результате анализа, определен перечень случайных ситуаций, которые могут произойти в процессе работы интернет-провайдера, и, соответственно, повлиять на прибыль. Результаты анализа представлены в таблице 1.

В данной задаче, на прибыль предприятия влияет не только перечень случайных ситуаций, но и набор услуг, которые может предоставить фирма. В связи с этим определен перечень основных услуг, предоставляемых интернет-провайдером:

- предоставление интернет услуг (подключение интернета с различными характеристиками физическим и юридическим лицам);
- переподключение абонентов от других интернет-провайдеров;
- диагностика оборудования клиента;
- вызов монтажника;
- настройка оборудования клиента (роутера);
- восстановление настроек сетевого оборудования.

Таблица 1. Перечень случайных ситуаций, влияющих на прибыль предприятия

| Случайная ситуация | Описание |
|-------------------------------------|---|
| Погодные условия | Природные факторы, которые могут приводить к поломкам оборудования провайдера и клиента. Можно выделить три погодных условия, влияющих на работу интернет-провайдера: гроза, при которой могут выйти из строя коммутаторы и кабель; при сильном ветре, урагане повышается вероятность падения деревьев, в результате чего может быть повреждено оборудование; аномальная жара, при которой оборудование перегревается и выходит из строя. |
| Сбой оборудования клиента | К оборудованию клиента относятся: роутер, компьютер, сетевой адаптер. Так же на стороне клиента могут произойти: сбой параметров системы, сбой настроек файервола, заражение вирусами, блокирующими работу интернета. Кроме этого могут выйти из строя коммуникации клиента: кабель, интернет-розетки. |
| Сбой работы оборудования провайдера | К оборудованию провайдера относят коммутаторы, кабель, сервер. При выходе из строя данного оборудования, предприятие может понести большие потери. |
| Количество абонентов | От количества абонентов напрямую зависит прибыль предприятия. Нельзя назвать конкретное число абонентов, при котором предприятие будет получать доход, так как большую роль играет регион, где будет расположен интернет-провайдер, его населенность, а также качество услуг, которые предоставляют фирмы-конкуренты в данном регионе. |

В теории принятия решений очень важным является определение типа решаемой задачи. Выяснение этого вопроса влияет на объем и структуру необходимой информации, а также на выбор методов, используемых при решении задачи. Из формулировки цели данной задачи можно сразу ограничиться решением на основе одного критерия, определяющего целесообразность функционирования предприятия. Из теории известно, что в зависимости от доступной информации, задачу можно отнести к одной из трех типов [2]:

- в условиях определенности;
- в условиях риска;
- в условиях неопределенности.

Для рассматриваемой задачи отсутствует вся необходимая информация, поэтому можно рассматривать только второй и третий вариант. Кроме этого, такие задачи для формализации требуют получения следующей информации:

- определения множества всех возможных случайных событий, влияющих на функционирование предприятия, а значит, на получение возможного дохода;
- определение множества всех доступных видов услуг, которые может предоставить предприятие;
- оценка этих видов услуг с позиции получаемого дохода.

Структурированная таким образом информация служит для дальнейшего анализа задачи: если известны вероятности случайных событий или они могут быть определены по статистическим данным и при этом лицо, принимающее решение доверяет этой вероятности, тогда задача должна быть формализована как задача в условиях риска [3] и решаться соответствующими методами, в противном случае решение принимается в условиях неопределенности.

В данной задаче, одним из определяющих факторов является внешняя среда, т. е. перечень случайных ситуаций, которые могут находиться в одном из k состояний S_1, S_2, \dots, S_k , вероятность возникновения которых неизвестна лицу, принимающему решение. Это значит, что мы не можем точно определить, будет ли гроза и приведет ли она к поломке оборудования. В основном, такие данные можно получить статистическим путем или же с помощью экспертов в данной области [4]. Но если необходимые для принятия решений статистические данные отсутствуют (как в рассматриваемой задаче), или имеется недоверие к таким данным, то задачи указанного типа относят к задачам принятия решений в условиях неопределенности [4].

Для формализации задачи в условиях неопределенности необходимо выполнить следующие действия:

1. Определить множество $\{Q_1, Q_2, \dots, Q_n\}$ всех возможных внешних ситуаций, которые влияют на экономические результаты решения;
2. Составить перечень $\{X_1, X_2, \dots, X_n\}$ анализируемых альтернативных решений;
3. Определить модель представления задачи;
4. Определить ожидаемые доходы при принятии соответствующего решения;
5. Выбрать критерий для расчета наилучшего решения.

Определение множества возможных ситуаций, влияющих на экономические результаты решения

Указанный набор ситуаций $\{Q_j, j=\overline{1, n}\}$ должен представлять собой полную группу событий. Это означает, что должны выполняться следующие два условия, представленные в формуле (1).

$$\forall (k, l) \quad Q_k \cap Q_l = \emptyset \quad (1)$$

т.е. одновременное наступление любых двух событий такой полной группы – невозможно, как показано в формуле (2).

$$\bigcup_{j=1}^n Q_j = \Omega \quad (2)$$

т.е. одно из событий полной группы наступит обязательно.

При определении числа событий полной группы во множестве всех возможных ситуаций $\{Q_1, Q_2, \dots, Q_n\}$, которые необходимо учесть при анализе решений, удобно пользоваться основным правилом комбинаторики [6].

Сформулируем сначала это правило применительно к случаю, когда учитываются только два фактора, каждый из которых обуславливает свой специфический вид неопределенности, отражающийся на экономическом результате.

Например, пусть первый фактор - это погодные условия, а второй - сбой оборудования клиента. Для первого из них в модели требуется учесть n_1 различных сценариев развития событий, а для второго – n_2 . Тогда, согласно указанному правилу, при формализации соответствующей полной группы событий всего необходимо учесть $n_1 * n_2$ различных сценариев развития событий. Основное правило комбинаторики соответственно распространяется на случай произвольного числа учитываемых факторов.

В частности, для рассматриваемой задачи, требуется учесть следующие факторы, описанные в таблице 2.

Тогда применительно к данной задаче при формализации полной группы событий $\{Q_j\}$ необходимо учесть $4 * 2 * 2 * 3 = 48$ вариантов развития событий.

Рассмотрение выделенных случайных ситуаций во всех возможных комбинациях (т.е. 48 вариантов) является очень громоздкой задачей. В связи с этим, для примера рассмотрим только некоторые из них:

- Q1 - ситуация, при которой {Погодные условия - благоприятные} * {сбой оборудования клиента – не произошел} * {Сбой работы оборудования провайдера - не произошел} * {Количество абонентов - много};

- Q2 - ситуация, при которой {Погодные условия - благоприятные} * {сбой оборудования клиента – не произошел} * {Сбой работы оборудования провайдера - не произошел} * {Количество абонентов - среднее количество};

- Q3 - ситуация, при которой {Погодные условия - благоприятные} * {сбой оборудования клиента – не произошел} * {Сбой работы оборудования провайдера - не произошел} * {Количество абонентов - клиентов много};

- Q4 - ситуация, при которой {Погодные условия – не благоприятные} * {сбой оборудования клиента – произошел} * {Сбой работы оборудования провайдера - не произошел} * {Количество абонентов - много};

- Q5 - ситуация, при которой {Погодные условия – не благоприятные} * {сбой оборудования клиента – не произошел} * {Сбой работы оборудования провайдера - произошел} * {Количество абонентов - много}.

Таблица 2. Факторы, влияющие на прибыль интернет-провайдера

| Фактор | Возможные сценарии развития событий |
|-------------------------------------|---|
| Погодные условия | гроза; сильный ветер; аномальная жара; погода, не влияющая на работу провайдера. |
| Сбой оборудования клиента | сбой произошел; сбой не произошел. |
| Сбой работы оборудования провайдера | сбой произошел; сбой не произошел. |
| Количество абонентов | абонентов – мало; среднее количество абонентов; абонентов – много. |

Составление перечня альтернативных решений

Альтернативное решение представляет собой набор действий (в нашем случае работ или услуг), выполнение которых приносит прибыль. Составление перечня всех возможных альтернативных решений, влияющих на экономический результат, является сложной задачей, так как альтернатив может быть очень большое количество. Но вопрос о том, рассматривать или нет в рамках соответствующего анализа конкретную альтернативу, решает непосредственно лицо, принимающее решение. В связи с этим выделены основные альтернативные решения:

X_1 – Предоставление интернет-услуг;
 X_2 – Предоставление интернет-услуг, переподключение абонентов от других интернет-провайдеров;

X_3 – Предоставление интернет-услуг, переподключение абонентов от других интернет-провайдеров, диагностика оборудования клиента, вызов монтажника, настройка оборудования клиента, Восстановление настроек сетевого оборудования.

X_4 – Предоставление интернет-услуг, переподключение абонентов от других интернет-провайдеров, диагностика оборудования клиента, вызов монтажника;

X_5 – Предоставление интернет-услуг, переподключение абонентов от других интернет-провайдеров, диагностика оборудования клиента, вызов монтажника, настройка оборудования клиента;

X_6 – Предоставление интернет-услуг, переподключение абонентов от других интернет-провайдеров, диагностика оборудования клиента;

Экономический результат альтернативных решений будет зависеть от того, какое из событий полной группы случайных ситуаций $\{Q_j, j=\overline{1, n}\}$, выделенной на предыдущем шаге, наступит.

Определение модели представления задачи

Получив всю необходимую информацию выбираем модель представления задачи:

- в виде матрицы доходов (полезностей) [7];

- в виде матричной игры [8], где игрок А – предприятие, а игрок В - окружающая среда; стратегии игрока А – это виды услуг, а стратегии игрока В – случайные события, влияющие на доход предприятия.

Обе модели имеют свои плюсы и минусы. Рассмотрим их подробнее.

Первая модель – это матрица, строки которой – все возможные комбинации услуг,

представляющих альтернативные варианты деятельности предприятия, а столбцы – полная группа случайных событий. Для такой модели матрица полезности имеет следующий вид, представленный в таблице 3.

В приведенной матрице значения $\{X_1, X_2, \dots, X_m\}$ характеризуют каждый из вариантов альтернатив принятия решения. Значения $\{Q_1, Q_2, \dots, Q_n\}$ – каждый из возможных вариантов ситуации развития событий, а значения $\{A_{11}, A_{12}, \dots, A_{nm}\}$ - конкретный уровень эффективности решения (доход), соответствующий определенной альтернативе при определенной случайной ситуации.

Таблица 3. Матрица полезности в общем виде

| Варианты альтернативных решений | Варианты ситуаций развития событий | | | | |
|---------------------------------|------------------------------------|----------|----------|-----|----------|
| | Q_1 | Q_2 | Q_3 | ... | Q_n |
| X_1 | A_{11} | A_{12} | A_{13} | ... | A_{1n} |
| X_2 | A_{21} | A_{22} | A_{23} | ... | A_{2n} |
| X_3 | A_{31} | A_{32} | A_{33} | ... | A_{3n} |
| ... | ... | ... | ... | ... | ... |
| X_m | A_{m1} | A_{m2} | A_{m3} | ... | A_{mn} |

На основе указанной матрицы рассчитывается наилучшее из альтернативных решений по избранному критерию. Для решения данной задачи наиболее распространенными критериями принятия решений являются [9]:

- максиминный критерий (критерий Вальда);
- оптимистический критерий;
- нейтральный критерий;
- критерий Сэвиджа;
- производные критерии;

Следует отметить, что приведенные критерии являются далеко не единственными для принятия решения в условиях неопределенности. Однако, остальные критерии, в основном, являются комбинацией приведенных критериев и образуют группу составных критериев.

Таким образом, данная модель характеризуется громоздкостью, сложностью для оценки всех возможных ситуаций, но проста для окончательного принятия решения.

Вторая модель – это также матрица, но ее размерность определяется только количеством строк – это возможные виды услуг, и столбцов – это случайные ситуации. То есть размерность такой матрицы существенно меньше, чем в первой модели. Заполняется матрица также значениями возможного дохода от конкретной услуги в определенной ситуации, что не сложно, так как является обязательной необходимой

информацией, которая получена на этапе исследования задачи.

Проблема возникает на этапе анализа матрицы по доминируемым строкам. Многие виды деятельности предприятия, являющиеся обязательными, не всегда приносят доход, соизмеримый с другими менее обязательными видами услуг. Поэтому, прежде чем заполнять такую матрицу, следует решить проблему соизмеримости альтернатив.

Эта проблема существует и для первой модели. Ее решение заключается в нормировании матриц [10], то есть приведении элементов матрицы к безразмерным величинам (к единой шкале).

Для нормирования переменной A_{ij} , значение которой стремиться к максимуму, необходимо использовать формулу (3).

$$A_{ij} = \frac{A_{ij} - A_{jmin}}{A_{jmax} - A_{jmin}} \text{ при } C_j \rightarrow \max \quad (3)$$

Для нормирования переменной A_{ij} , значение которой стремиться к минимуму, необходимо использовать формулу (4).

$$A_{ij} = \frac{A_{jmax} - A_{ij}}{A_{jmax} - A_{jmin}} \text{ при } C_j \rightarrow \min \quad (4)$$

Следовательно, после нормирования матрицы ее значения будут лежать в диапазоне [0; 1]. При чем значение 0 соответствует худшему значению, а 1 – лучшему.

Решение задачи в виде матричной игры позволит получить смешанную стратегию игрока А (руководителя предприятия) в виде вероятностей использования альтернатив для получения максимального дохода. Полученные вероятности можно интерпретировать как процентное отношение различных видов услуг в общей деятельности предприятия.

Решение задачи относительно игрока В (внешние условия) позволит определить вероятности выбранных случайных событий, при которых доход предприятия будет наилучшим. Анализ полученных результатов можно использовать для оценки целесообразности открытия фирмы или расширения ее сферы деятельности.

Таким образом, вторая модель характеризуется малой размерностью матрицы, простотой ее заполнения, легкостью анализа альтернатив при различных событиях, но недостатком матричной игры является то, что она не учитывает наступление нескольких случайных событий одновременно, что в реальности может произойти и повлиять на прибыль предприятия.

Выводы

В результате исследования внешних и внутренних факторов, влияющих на работу интернет-провайдера, можно сделать вывод, что принятие решения о целесообразности открытия

или расширения сферы деятельности фирмы является очень трудоемкой задачей, так как для ее решения необходимо провести анализ случайных ситуаций, которых может быть сколь угодно много и которые могут принимать различное число состояний. Кроме этого нужно определить набор услуг, которые может предоставлять предприятие.

Не менее трудоемким является процесс построения матрицы полезности, предназначение которой - определить доход предприятия для каждой рассматриваемой альтернативы при влиянии возможных случайных ситуаций. Можно заметить, что чем больше факторов, тем больше размерность матрицы полезности и тем сложнее процесс выбора наилучшей альтернативы.

Рассмотрены две модели представления задачи: в виде матрицы полезностей и матричной игры. Первая модель является сложной для анализа всех ситуаций, но легкой для принятия решения. Вторая модель предназначена для анализа альтернатив при различных случайных ситуациях.

С увеличением размерности задачи (увеличение количества случайных ситуаций и их состояний, количества предлагаемых услуг и их комбинаций) является целесообразным разработать программную систему, с помощью которой можно решить проблему определения полной группы случайных событий, перебора всех возможных альтернативных решений, построения матрицы полезности и выбора метода решения полученной задачи.

Литература

1. Черноруцкий И.Г. Методы принятия решений. - СПб.:БХВ-Петербург, 2005. – 416 с.
2. Халин В.Г. Теория принятия решений: учебник и практикум для бакалавриата и магистратуры. – М.: Юрайт, 2016. – 250 с.
3. Петровский А.Б. Теория принятия решений: учебник для студ. высш. учеб. заведений. – М.: Академия, 2009. – 400 с.
4. Колпакова Т.А. Определение компетентности экспертов при принятии групповых решений // Радиоэлектроника, информатика, управление. – 2011. - №1 (24). – с.40-43.
5. Ларичев О.И. Теория и методы принятия решений, а также хроника событий в волшебных странах: учебник. - М.: Логос, 2000. – 296 с.
6. Бродецкий Г.Л. Системный анализ в логистике. Выбор в условиях неопределённости. – М.: Академия, 2010. – 336 с.
7. Евланов Л.Г. Теория и практика принятия решений. – М.: Экономика. 1984. – 176 с.
8. Гладких Б. А. Методы оптимизации и исследование операций для бакалавров

информатики. Ч. III. Теория решений: учебное пособие. – Томск: НТЛ, 2012. – 281 с.

9. Мушик Э., Мюллер П. Методы принятия технических решений: пер. с нем. – М.: Мир, 1990. – 208 с.

10. Афоничкин А.И., Михаленко Д.Г. Управленческие решения в экономических системах: учебник для вузов. – СПб.: Питер, 2009. – 480 с.

Матях И.В., Савкова Е.О., Ченгарь О.В. Исследование проблемы выбора минимального набора услуг для расширения деятельности предприятия на примере интернет-провайдера. В статье рассматривается формализация задачи принятия решения о целесообразности открытия или расширения сферы деятельности фирмы, на примере интернет-провайдера. Для этого исследованы внешние и внутренние факторы, влияющие на прибыль предприятия. Рассмотрены две модели представления задачи: в виде матрицы полезностей и матричной игры. Выделены достоинства и недостатки данных моделей, описаны методы решения задачи.

***Ключевые слова:** интернет-провайдер, прибыль, случайная ситуация, набор услуг, матрица полезностей, матричная игра, альтернативное решение, правило комбинаторики.*

Matiakh I.V., Savkova E.O., Chengar O.V. Investigation of the problem of choosing a minimum set of services for expanding the activity of an enterprise using the example of an Internet provider. The article deals with the formalization of the problem of making a decision about the expediency of opening or expanding the scope of the firm's activities, using the example of an Internet provider. For this, external and internal factors affecting the profit of the enterprise were investigated. Two models of the problem representation are considered: in the form of utility matrix and matrix game. The advantages and disadvantages of these models are singled out, methods for solving the problem are described.

***Keywords:** Internet provider, profit, random situation, set of services, utility matrix, matrix game, alternative solution, combinatorial rule.*

*Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко*

УДК 517.9

Приближенный анализ конвективной задачи Стефана

Миненко А.С., Радевич Е.В.

Донецкий национальный технический университет

radevich_katerina@mail.ru

Миненко А.С., Радевич Е.В. Приближенный анализ конвективной задачи Стефана. Осуществляется математическое моделирование одного класса сложных систем с применением нечеткой логики, а также численный анализ нелинейной математической модели.

Ключевые слова: задача Стефана, анализ, математическое моделирование, математическая модель, нечеткая логика.

Введение

Теплофизические процессы, сопровождающиеся фазовыми переходами вещества, описываются математической моделью, в которой температура каждой из фаз удовлетворяет уравнению переноса тепла со своими теплофизическими коэффициентами, на границе раздела фаз, обе температуры постоянны и равны температуре фазового перехода, а на заданных частях границы поддерживается определенный режим. Поверхность раздела фаз является неизвестной или «свободной» границей, и для ее определения дополнительно задается так называемое «условие Стефана», означающее, что тепловой поток через фронт кристаллизации в сторону твердой фазы равен тепловому потоку со стороны жидкой фазы плюс скрытая теплота фазового перехода.

Математическое моделирование одного класса сложных систем с применением нечеткой логики

Рассмотрим область $\Omega = \{(x_1, x_2, x_3) : r^2 < x_1^2 + x_2^2 + x_3^2 < R^2\}$ и через Γ^- и Γ^+ обозначим следующие сферы: $\Gamma^- = \{(x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 = r^2\}$, $\Gamma^+ = \{(x_1, x_2, x_3) : x_1^2 + x_2^2 + x_3^2 = R^2\}$. Далее, пусть Γ_0 гладкая, связная поверхность без самопересечений, лежащая внутри Ω , которая разбивает ее на две подобласти Ω^+ и Ω^- , т.е. $\Omega = \Omega^+ \cup \Omega^-$, причем сфера Γ^- лежит внутри ограниченной области, границей которой является Γ_0 . Рассмотрим краевую задачу со свободной границей Γ_0 . Требуется определить тройку $(u^\pm(x), \Gamma_0)$ по следующим условиям:

$$\begin{aligned} \nabla^2 u^\pm(x) &= 0, x \in \Omega^\pm; u^\pm(x)|_{\Gamma^\pm} = B^\pm(x); \\ u^\pm(x) &= 1, |\nabla u^-(x)| - |\nabla u^+(x)| = 0, x \in \Gamma_0. \end{aligned} \quad (1)$$

При этом,

$$B^\pm(x) \in C^{3+\alpha}(\Gamma^\pm), u^\pm(x) \in C^{3+\alpha}(\overline{\Omega^\pm}), \quad \text{а } \Gamma_0 \text{ принадлежит классу } C^\infty [12].$$

Затем введем в рассмотрение функцию $u(x)$, заданную следующим образом $u = u^-(x)$, при $x \in \overline{\Omega^-}$ и $u = u^+(x)$, при $x \in \overline{\Omega^+}$. Тогда функцию $u(x)$ можно найти из условия минимума функционала

$$I(u, \Gamma_0) = \iint_{\Omega} |\nabla u|^2 dx_1 dx_2 dx_3 \text{ на соответствующем}$$

множестве R допустимых функций [19]. Это следует из формулы первой вариации интегрального функционала с неизвестной областью интегрирования [6].

Далее, удобно представить функционал I в сферических координатах:

$$I(u, \Gamma_0) = \int_0^{2\pi} \int_0^\pi \int_r^R \left(u_\rho^2 + \frac{1}{\rho^2} u_\theta^2 + \frac{1}{\rho^2 \sin^2 \theta} u_\phi^2 \right) \rho^2 \sin \theta d\phi d\theta d\rho \quad (2)$$

Лемма 4.1. Пусть тройка $(u^\pm(x), \Gamma_0)$ является классическим решением задачи (4.1). Тогда эта тройка будет стационарной для функционала (4.2) на множестве R . Обратно, каждая стационарная тройка $(u^\pm(x), \Gamma_0)$ функционала (4.2) на множестве R , где Γ_0 – достаточно гладкая, связная поверхность, является решением задачи (4.1).

Сформулированная задача (4.1) получается из задачи, изученной в [12] в случае $\vec{V} = 0$, т.е. в случае бесконечно большой вязкости, $Re = 0$. Поэтому в дальнейшем под решением задачи (4.1) при $Re = 0$ будем понимать функции $\vec{V}(x) = 0, u^+(x)$ и $u^-(x)$, заданные в Ω^\pm . Из условий (4.1) следует, что Γ_0 – не что иное, как линия уровня функции $u(x)$, т.е.:

$\Gamma_0 = \{x \in \Omega : u(x) = 1\}$. Далее, если предположить выполнение следующего условия:

$$\pm(B^\pm(x) - 1) \geq \varepsilon_0 > 0, x \in \Gamma^\pm,$$

где ε_0 – некоторая постоянная, тогда поверхность Γ_0 лежит внутри области Ω и представляет собой поверхность класса $C^{4+\alpha}$, не имеющую самопересечений и располагающую относительно Γ^+ и Γ^- аналогично поверхности Γ_t (свободная поверхность), изученной в [12]. Следовательно, рассматривая функцию $u(x)$ в одной из областей Ω^\pm , и принимая во внимание лемму о нормальной производной, находим что:

$$\frac{\partial u}{\partial n} = |\nabla u| \geq \varepsilon > 0, x \in \Gamma_0,$$

где n – нормаль к Γ_0 , направленная в сторону Ω_0^+ , а ε – некоторая постоянная. Отсюда, применяя теорему о неявной функции, следует, что Γ_0 принадлежит классу C^∞ , так как этому классу в некоторой окрестности Γ_0 принадлежит гармоническая функция $u(x)$.

Минимум функционала (4.2) на множестве R будем искать при помощи сумм:

$$u_n = B^+ + \frac{R^2 - \rho^2}{R^2 - r^2}(B^- - B^+) + (R^2 - \rho^2)(\rho^2 - r^2) \sum_{k=0}^n C_k \rho^k y_k(\varphi, \theta),$$

где $y_k(\varphi, \theta)$ – сферические функции. Неизвестные коэффициенты C_k определяют при помощи метода Ритца. Тогда поверхность $\Gamma_0 : \rho = \rho_0(\varphi, \theta)$ определяется из уравнения $u_n(\varphi, \theta, \rho_0(\varphi, \theta)) = 1$.

При этом, необходимо учесть, что $|\nabla u(x)| \geq \varepsilon_0 > 0$, в $\bar{\Omega}$, где ε_0 – некоторая постоянная [3].

Лемма 4.2. При малых t справедливо представление:

$$\Gamma_t : \rho(\varphi, \theta, t) = \rho_0(\varphi, \theta) - \text{Re} \frac{u_1^\pm(\varphi, \theta, t)}{|\nabla A^\pm(\varphi, \theta)|} + 0(\text{Re}), (\varphi, \theta) \in \Gamma_0. \quad (3)$$

Здесь Re – число Рейнольдса, а $u_1^\pm(\varphi, \theta, t)$ – первое приближение исходной задачи, изученной в [12].

В частности для нулевого приближения $u_0(\varphi, \theta)$ из уравнения:

$$u_0 = B^+ + \frac{R^2 - \rho^2}{R^2 - r^2}(B^- - B^+) + (\rho^2 - r^2)(R^2 - \rho^2)C_0 = 1$$

легко найти поверхность $\rho_0(\varphi, \theta)$.

Далее, рассмотрим величину, $\varepsilon_n = I(u_n, \Gamma_0) - I(u, \Gamma_0)$, где u – точное решение задачи (4.1). Тогда, можно установить, что $\varepsilon_n \rightarrow 0$ при $n \rightarrow \infty$, если C_k – коэффициенты Ритца.

Используя результаты Канторовича Л.В. по минимизации квадратичных функционалов, аналогично тому как это сделано в [13], можно доказать следующее утверждение.

Теорема 4.1. Последовательность приближений Ритца u_n сходится к решению задачи (1) u по норме в W_2^1 и C , причем $\varepsilon_n = O\left(\omega^{(3)}\left(u, \frac{1}{n}\right) / n^2\right)$, и если

$$\lim_{n \rightarrow \infty} \omega^{(3)}\left(u, \frac{1}{n}\right) n^{-1} (\ln n)^{1+\varepsilon} = 0, \quad \text{тогда:}$$

$$\|u - u_n\|_{C(\bar{\Omega})} \leq C_1 \omega^{(3)}\left(u, \frac{1}{n}\right) \cdot n^{-1} + C_2 \sum_{s=m}^{\infty} \omega^{(3)}\left(u, \frac{1}{2^s}\right) \cdot 2^{-s},$$

где C_1 и C_2 – некоторые постоянные, $\omega^{(3)}\left(u, \frac{1}{n}\right)$ – максимальный модуль непрерывности производных третьего порядка функции $u(x)$ и $2^{m-1} \leq m < 2^m$.

Замечание. В случае двух геометрических переменных имеют место оценки:

$$\varepsilon_n = O\left(\frac{1}{n^{2(2+\alpha)}}\right), \|u_n - u\|_{C(\bar{\Omega})} \leq C_1 \sqrt{\varepsilon_n \ln \frac{n}{\varepsilon_n}} + C_2 \sqrt{\varepsilon_n}, \quad (4)$$

В работе [10] изучены k -е приближения $(\vec{V}, u_k^\pm, \rho_k)$ исходной задачи являющиеся функциями класса $H^{2+\alpha, (2+\alpha)/2}(\bar{\Omega}_2^\pm)$, построены системы уравнений, решениями которых они являются. Формулы (4.3), (4.4) позволяют исследовать Γ_t в зависимости от чисел Рейнольдса Re .

Пусть T^* – температура, которую должна достичь поверхность $\partial\Omega$. Эта температура достигается за счет воздействия тепловых потоков мощности w_1, w_2, w_3 , причем мощность одного из них w_3 равномерно распределена в центре $\partial\Omega$, а два других w_1 и w_2 сконцентрированы по краям $\partial\Omega$ [17]. Далее, будет предложен метод нечеткого управления в данном классе задач, который имеет место в спецметаллургии [17].

Пусть X_1, X_2, \dots, X_n – факторы, влияющие на процессе кристаллизации, а Y_1, Y_2, \dots, Y_n – условия, при которых происходит появление нового слитка. Тогда нечеткое управление в нашей модели можно представить в виде функционального отображения: $X = \{X_1, X_2, \dots, X_n\} \rightarrow Y = \{Y_1, Y_2, \dots, Y_n\}$.

В простейшем случае, например, в качестве терм-множества лингвистических переменных x_1, x_2, x_3 , где $x_1 = \{\text{“температура слитка”}\}$, $x_2 = \{\text{“способ нагрева”}\}$, $x_3 = \{\text{“слиток металла”}\}$ можно использовать

соответственно множества: $T = \{\text{“минимальная”}, \text{“средняя”}, \text{“максимальная”}\}$, $W = \{\text{“минимальный”}, \text{“средний”}, \text{“максимальный”}\}$, $L = \{\text{“минимальный”}, \text{“средний”}, \text{“максимальный”}\}$. Следовательно, получим:

$$x = \{x_1, x_2, x_3\} \rightarrow y \in [a, b],$$

где a и b – некоторые числа, а для выходной лингвистической переменной y (температура поверхности слитка) будет использоваться термножество $Q = \{\text{“минимальная”}, \text{“средняя”}, \text{“максимальная”}\}$. Пределы a и b выбираются таким образом, чтобы произошло отделения слитка от стенок кристаллизации [17]. Далее, формируется база нечетких высказываний из 17 правил.

При численной реализации задачи использовались следующие значения параметров:

$$2500 \text{ MВт} / \text{м}^2 \leq W \leq 5000 \text{ MВт} / \text{м}^2, 600 \text{ мм} \leq L \leq 6000 \text{ мм}.$$

Численный расчет, позволяющий построить нечеткое управление, был осуществлен с помощью стандартного алгоритма Мамдани, а результаты получены в ходе эксперимента на объектах управления ЭСП [17].

Численный анализ одной нелинейной математической модели

Пусть $\Omega \in R^3$ – заданная область, граница которой $\partial\Omega$ состоит из двух замкнутых, связных гладких поверхностей Γ^+ и Γ^- , не имеющих самопересечений, причем поверхности Γ^\pm предполагаются принадлежащими классу $H^{5+\alpha}$, $0 < \alpha < 1$. Пусть далее $\Gamma_t(t \in [0, T])$ – гладкие замкнутые поверхности, лежащие внутри Ω , такие, что Γ^+ лежит внутри ограничений области, границей которой является Γ_t . Свободная поверхность Γ_t – граница раздела фаз в момент времени t – разбивает область Ω на две связные подобласти Ω_t^- и Ω_t^+ , занимаемых твердой и жидкой фазами соответственно.

Требуется определить вектор скорости $\vec{V}(x, t)$, давление $p(x, t)$, распределения температур твердой и жидкой фаз $u^-(x, t)$ и $u^+(x, t)$ и свободную поверхность Γ_t по следующим условиям:

$$\frac{\partial \vec{V}}{\partial t} + (\vec{V} \nabla) \vec{V}(x, t) + \nabla p(x, t) = \frac{1}{\text{Re}} \nabla^2 \vec{V}(x, t) + \vec{f}(u^+), \quad (5)$$

$$\nabla \vec{V}(x, t) = 0, (x, t) \in D_T^+;$$

$$\frac{\partial}{\partial t} u^+(x, t) + (\vec{V} \nabla) u^+(x, t) - a_+^2 \nabla^2 u^+(x, t) = 0, (x, t) \in D_T^+, \quad (6)$$

$$\frac{\partial}{\partial t} u^-(x, t) - a_-^2 \nabla^2 u^-(x, t) = 0, (x, t) \in D_T^-, \quad (7)$$

$$u^\pm(x, t) \Big|_{t=0} = A^\pm(x), u^\pm(x, t) \Big|_{x \in \Gamma^+ \cup \Gamma^-} = B^\pm(x, t), \quad (8)$$

$$\vec{V}(x, t) \Big|_{t=0} = \vec{C}(x), \vec{V}(x, t) \Big|_{x \in \Gamma^+ \cup \Gamma_t} = 0, \quad (9)$$

$$u^\pm(x, t) \Big|_{x \in \Gamma_t} = 0, \sum_{i=1}^3 \left[K_- \frac{\partial u^-}{\partial x_i} - K_+ \frac{\partial u^+}{\partial x_i} \right] \cos(n, x_i) + K \cos(n, t) = 0, x \in \Gamma_t, \quad (10)$$

где $D_T^\pm = \{(x, t) : x \in \Omega_t^\pm, t \in (0, T)\}$, $\partial\Omega^\pm \Gamma_t \cup \Gamma^\pm$, $\nabla = (\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3)$, \vec{n} – нормаль к Γ_t , направлена в сторону Ω_t . Предполагается, что $B^\pm(x, t) \in H^{3+\beta, (3+\beta)/2}(\Gamma^\pm \times [0, T])$, $0 < \beta < \alpha$, $A^\pm(x) \in H^{5+\alpha}(\overline{\Omega_0^\pm})$, $\vec{C}(x) \in H^{2+\alpha}(\overline{\Omega^+})$, где Ω_0^\pm – области, на которые разбивает Ω граница раздела фаз Γ_0 в момент времени $t=0$ и $B^\pm(x, t) \geq \varepsilon_0 > 0$ при $(x, t) \in \Gamma^\pm \times [0, T]$.

Параметры a_\pm , K_\pm , K , Re , ε_0 – считаются положительными постоянными, а $\vec{f}(u^+)$ – принадлежащей классу $C^2(R^1) f'(u^+)$ – ограниченной в R^1 . Задача (4.5) –(4.10) при малых значениях t разрешима в классе гладких функций, при этом $u^\pm \in H^{2+\alpha, (2+\alpha)/2}(\overline{D_T^\pm})$, $\vec{V} \in H^{2+\beta, (2+\beta)/2}(\overline{D_T^\pm})$, а свободная поверхность Γ_t принадлежит классу $H^{2+\alpha, (2+\alpha)/2}$ [5].

Настоящая работа посвящена приближенному анализу задачи (4.5) –(4.10).

Для точек поверхности Γ_0 введем координаты $\omega = (\omega_1, \omega_2)$, через $x(\omega) \in \Gamma_0$ или через ω будем обозначать также соответствующие точки в R^3 . Далее, пусть $\vec{n}(\omega)$ – нормаль к Γ_0 , направленная внутрь Ω_0^+ . В работе [10] установлено, что поверхность Γ_t можно представить в виде $\Gamma_t = \left\{ x = x(\omega) + \vec{n}(\omega) \rho(\omega, t) \right\}$ с некоторой функцией $\rho(\omega, t)$ класса $H^{2+\alpha, (2+\alpha)/2}(\Gamma_0 \times [0, T])$, так что $\rho(\omega, 0) = 0$.

Предположим, что при малых значениях Re неизвестные нашей задачи можно представить в виде степенного ряда:

$$u^\pm(x, t) = u_0^\pm(x) +;$$

$$V_i(x, t) = V_{i0}(x) + \sum_{K=1}^{\infty} (\text{Re})^K V_{iK}(x, t), i = 1, 2, 3;$$

$$\rho(\omega, t) = \sum_{K=1}^{\infty} (\text{Re})^K \rho_K(\omega, t).$$

В работах [8-10] изучены нулевые и первые приближения задачи (4.5) – (4.10) для малых чисел Re . При этом установлено, что $u_0^\pm = A^\pm(x)$, $\vec{V}_0(x) = \vec{C}(x)$, $\rho_1(\omega, t) \in H^{2+\alpha, (2+\alpha)/2}(\Gamma_0 \times [0, T])$, $u_1^\pm(x, t) \in H^{2+\alpha, (2+\alpha)/2}(\overline{D_T^\pm})$, причем $\rho_1(\omega, t)$ находим как неподвижную точку сжимающегося оператора M_1 :

$$M_1 \rho_1 = \frac{1}{K} \int_0^t \left(K_- \frac{\partial u_1^-}{\partial n} - K_+ \frac{\partial u_1^+}{\partial n} + f(x, t) \right) dt,$$

$$x(\omega) \in [0, T],$$

а $f_1(x, t)$ - некоторая функция класса $H^{2+\alpha, (2+\alpha)/2}$

Рассмотрим случай, когда $B^\pm = B^\pm(x)$ и $\Omega = \{(x_1, x_2, x_3) : r^2 < x_1^2 + x_2^2 + x_3^2 < R^2\}$.

Тогда нулевое приближение находим как решение следующей задачи:

$$\begin{cases} \nabla^2 u^\pm(x) = 0, x \in \Omega_0^\pm, A^\pm(x)|_{\Gamma_0^\pm} = B^\pm(x), u^\pm(x)|_{\Gamma_0^\pm} = 0, \\ \vec{C}(x) = 0, x \in \overline{\Omega_0^\pm}, |\nabla u^-(x)| = |\nabla u^+(x)| = 0, x \in \Gamma_0 \end{cases} \quad (11)$$

Заметим, что замена $\tilde{u}^- = K_- u^-$ при $x \in \Omega^-$ и $\tilde{u}^+ = K_+ u^+$, если $x \in \Omega^+$ сводит задачу (4.11) к случаю $|\nabla u^-(x)| = |\nabla u^+(x)|, x \in \Gamma_0$. поэтому в дальнейшем будем считать, что это условие выполнено.

Нулевое приближение $u_0^\pm(x)$, Γ_0 найдем из условия минимума функционала

$$Y(u_0^\pm, \Gamma_0) = \iiint_{\Omega} |\nabla u|^2 dx_1 dx_2 dx_3, \quad \text{здесь}$$

$\Omega = \Omega_0^+ \cup \Omega_0^-$ и $u = u^-$ при $x \in \Omega^-$ и $u = u^+$, если $x \in \Omega^+$.

Далее, рассматривая функционал Y в сферических координатах, получим

$$Y(u_0) = \int_0^{2\pi} \int_0^\pi \int_0^R \left(u_\rho^2 + \frac{1}{\rho^2} u_\theta^2 + \frac{1}{\rho^2 \sin^2 \theta} u_\varphi^2 \right) \rho^2 \sin \theta d\varphi d\theta d\rho$$

Минимум функционала ищем в следующем виде

$$u = B^+ + \frac{R^2 - \rho^2}{R^2 - r^2} (B^- + B^+) + (R^2 - \rho^2)(\rho^2 - r^2) \sum_{K=0}^{\infty} C_K \rho^K y_c(\varphi, \theta)$$

Неизвестные коэффициенты C_K определяются методом Ритца. В частности, в случае нулевого приближения

$$u_0 = B^+ + \frac{R^2 - \rho^2}{R^2 - r^2} (B^- + B^+) + (R^2 - \rho^2)(\rho^2 - r^2) C_0,$$

из уравнения $\partial Y(u_0) / \partial C_0 = 0$ определим коэффициент C_0 . Справедлива следующая теорема.

Теорема 4.2. Поверхность Γ_0

представляет собой поверхность класса C^∞ , не имеющую самопересечений и расположенную относительно Γ^+ и Γ^- аналогично поверхности Γ_t в задаче (4.5) – (4.10).

Доказательство следует из принципа максимума, применённого к гармонической

функции $\Psi(x) = -\frac{\partial u_0(x)}{\partial \vec{r}}$ оценок

$$-\frac{\partial u_0(x)}{\partial \vec{r}} \Big|_{\overline{\Omega}} \geq \tilde{\varepsilon}_0 > 0 \quad \text{и теоремы о неявной}$$

функции, применённой к $\Psi(x)$. Здесь \vec{r} – радиус-вектор точки x .

Отсюда следует, что поверхность $\Gamma_0 : \rho = \rho_0(\varphi, \theta)$ можно найти из условия $u_0(\varphi, \theta, \rho_0(\varphi, \theta)) = 0$. Тогда для поверхности Γ_t можно воспользоваться уравнением [10]:

$$\Gamma_t = \rho(\varphi, \theta, t) = \rho_0(\varphi, \theta) - \text{Re} \frac{u_1^+(\varphi, \theta, t)}{|\nabla A^+(\varphi, \theta)|} + o(\text{Re})$$

На рис. 1 представлена поверхность Γ_t .

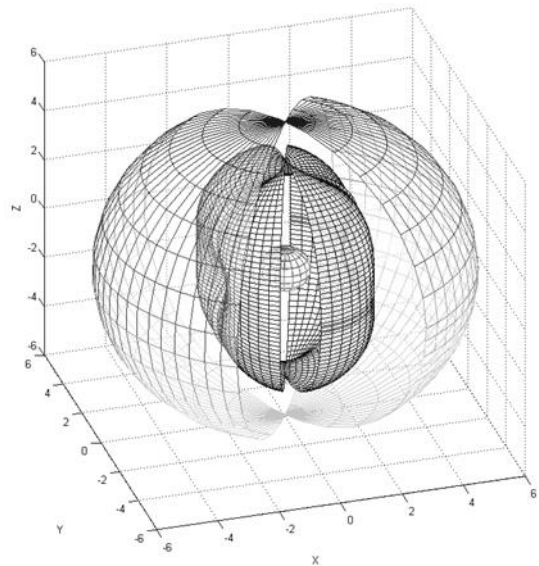


Рисунок 1 - Поверхность Γ_t

При построении использованы следующие значения параметров:

$$t = 200, R = 6, r = 0,8, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{3},$$

$$-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}, B^+ = 3[\cos^2 \theta + \cos^2 \varphi],$$

$$B = -0,35[\cos^2 \theta + \cos^2 \varphi] - 0,1.$$

Свободная поверхность Γ_t расположена между сферами радиусов R и r .

Предложенный алгоритм построения поверхности Γ_t позволяет исследовать эту поверхность в зависимости от параметров задачи (5) – (10).

Заключение

В заключение отметим, что в данной статье была рассмотрена задача Стефана для моделирования сложных систем.

Литература

1. Миненко А.С. Исследование одной конвективной задачи Стефана методом Ритца // Укр. мат. журнал. 2007. – 59, № 11. – С. 1546 – 1556.
2. Миненко А.С., Шевченко А.И. Об одной проблеме Стефана // Доповіді НАН України. – 2008. – № 1 – С. 26 – 30.
3. Миненко А.С. Проблема минимума со свободной границей // Искусственный интеллект. – 1998. – №2. – С. 101 – 109.
4. Данилюк И.И., Миненко А.С. Об одном подходе к анализу стационарной задачи Стефана при наличии конвекции в жидкой фазе. – В кн.: Мат. физика и Нелин.механика.- Киев, Наукова думка, 1985.- Вып. 65. – С. 39-48.
5. Миненко А.С., Шевченко А.И. Приближенный анализ стационарной конвективной задачи Стефана // Доповіді НАН України.-2010.-№5.- С.36-40.
6. Миненко А.С., Шевченко А.И. Приближенный анализ одной пространственной конвективной задачи теплопроводности // Доповіді НАН України. – 2007. – № 7. – С. 22 – 27.
7. Ладыженская О.А. Краевые задачи математической физики.-М.: Наука, 1973.- С.407.
8. Миненко А.С., Шевченко А.И. Исследование конвективного теплопереноса в одной пространственной задаче теплопроводности // Доповіді НАН України.- 2007. – № 9. – С.25-29.
9. Ладыженская О.А., Солонников В.А., Уральцева Н.Н. Линейные уравнения параболического типа. – М.: Наука, 1967. – 736 с
10. Миненко А.С., Шевченко А.И. Приближенный анализ многомерной конвективной задачи Стефана // Доповіді НАН України.- 2010.-№4 – С.30-34.
11. Данилюк И.И. О задачах Стефана // Успехи мат. наук. – 1985. – Т. 40, № 5(245). – С. 133 – 185.

12. Миненко А.С., Шевченко А.И. Математическое моделирование процессов кристаллизации металла с учетом конвекции и примесей //Доповіді НАН України.-2011.-№6- С.35-39.
13. Солонников В.А. разрешимость задачи о движении вязкой несжимаемой жидкости, ограниченной свободной поверхностью // Изв. АН СССР. Сер. мат.-1977.-41, №6.-С.1388-1424.
14. Данилюк И.И. Математическое моделирование фазовых превращений в двухкомпонентных средах //Доповіді НАН України.-1984.-№12.-С.10-13.
15. Лихт М.К., Кузьминская С.Б. О затвердевании бинарных сплавов в интервале температур. – Физика металлов и металловедения. – 1961. – 11, №6. – С.878-882.
16. Борисов В.Т. Кристаллизация бинарного сплава при сохранении устойчивости // Докл. АН СССР. – 1961, - 136, №3. – С.583-586.
17. Скворцов А.А. К решению вопроса о затвердевании металлов в интервале температур // Затвердевание металлов.-М.: Машиз.-1958.- С.124-160.
18. Борисов В.Т., Виноградов В.В., Духин А.И. О применимости теории квазиравновесной двухфазной зоны к описанию кристаллизации слитка // Изв. АН СССР. Сер. металлы.-1971, №6. – С.104-109.
19. Данилюк И.И. О смешанной задаче для квазилинейного управления теплопроводности с кусочно-разрывными коэффициентами // Докл. АН УССР. Сер. А. – 1981, №7. – С.3-7.
20. Данилюк И.И. О начально-краевой задаче для квазилинейного уравнения теплопроводности с кусочно-непрерывными коэффициентами // Проблемы математики и механики. – Новосибирск – 1983. – С.81-94.
21. Данилюк И.И., Миненко А.С. О методе Ритца в одной нелинейной задаче со свободной границей // Докл. АН УССР. Сер А. – 1978. – № 4. – С. 291 – 294.
22. Данилюк И.И., Миненко А.С. Об одной вариационной теплофизической задаче со свободной границей // Сб. докладов на конф. По смешанным граничным задачам и задачам со свободными границами. – Штутгарт, 1978. – С. 9 – 18.
23. Данилюк И.И., Миненко А.С. О вариационном методе изучения квазистационарной задачи Стефана // Успехи мат. наук. – 1981. – Т. 43, № 5. – С. 228.

Миненко А.С., Радевич Е.В. Приближенный анализ конвективной задачи Стефана. Осуществляется математическое моделирование одного класса сложных систем с применением нечеткой логики, а также численный анализ нелинейной математической модели.

Ключевые слова: задача Стефана, анализ, математическое моделирование, математическая модель, нечеткая логика.

Minenko A. S., Radevich E. V. Approximate analysis of convective Stefan problem. Performed mathematical modeling for a class of complex systems using fuzzy logic, and numerical analysis of nonlinear mathematical models.

Key words: Stefan problem, analysis, mathematical modeling, mathematical model, fuzzy logic.

*Статья поступила в редакцию 20.09.2017
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом*

Обучающая система работы с контейнерными классами

А.И. Ольшевский

Донецкий национальный технический университет
a_olshevskiy@mail.ru

Ольшевский А.И. Обучающая система работы с контейнерными классами. Рассмотрена классификация по типам основных структур данных контейнерных классов. Описаны возможности реализации их на языке Java с помощью коллекций, а в C++ – библиотекой STL. Определен подход формирования поддерживающий разные наборы операций, среди которых есть совпадающие операции. Реализован простой и понятный в использовании пользовательский интерфейс с подсказками и справкой по теоретической части для каждого типа контейнерных классов, а также примерами программ для конкретной реализации. Проанализирована эффективность разных методов соответствующих операций.

Ключевые слова: структуры данных, контейнерные классы, наборы операций, эффективность разных методов.

Введение

Компьютеризация общества и его различных институтов должна быть направлена не только (не столько) на пассивную (статическую) актуализацию информационных ресурсов общества, но и на создание и актуализацию новых информационных технологий, ресурсов, динамическое их переупорядочивание, а также их взаимопроникновение и взаимообогащение в системах вида «человек–технология–знания–система».

Применение контейнеров обусловливается возросшими объемами обрабатываемой информации. Когда счет используемых объектов идет на сотни тысяч, массивы не обеспечивают ни должной скорости, ни экономии ресурсов.

Но контейнерные классы – довольно сложная структура. Для того, чтобы в ней разобраться и изучить целесообразней использовать обучающую систему, созданную для этого.

Постановка задачи

Для каждого типа контейнера определены методы для работы с его элементами, не зависящие от конкретного типа данных, которые хранятся в контейнере, поэтому один и тот же вид контейнера можно использовать для хранения данных различных типов. В Java эта возможность реализована с помощью коллекций, а в C++ – стандартной библиотекой шаблонов STL (Standard Template Library).

Коллекции – это хранилища, поддерживающие различные способы накопления и упорядочения объектов с целью

обеспечения возможностей эффективного доступа к ним. Они представляют собой реализацию абстрактных типов (структур) данных, поддерживающих три основные операции: добавление нового элемента в коллекцию, удаление элемента из коллекции, изменение элемента в коллекции. В качестве других операций могут быть реализованы следующие: просмотреть элементы, подсчитать их количество и др.

Библиотека STL – это набор шаблонных классов и функций общего назначения. Но основное внимание стоит уделить контейнерам. Контейнеры – это объекты, содержащие другие объекты. Существует несколько различных типов контейнеров, которые реализуют различные алгоритмы обработки содержащихся в них данных [1].

Объектом разработки является коллекций языка Java и шаблоны классов, часть библиотеки C++, в которую входят контейнерные классы, а также алгоритмы и итераторы.

Цель проекта – создание программного обеспечения для повышения эффективного обучения пользователей возможностям коллекций, содержащих контейнеры, реализующие основные структуры данных, проектирования и написания программ, за счет использования мультимедийных технологий. Использование контейнеров позволяет значительно повысить надежность программ, их переносимость и универсальность, а также уменьшить сроки их разработки.

Разрабатываемый пакет программ позволит уменьшить время обучения и упростить затраты на вдумчивое освоение библиотеки для эффективного использования контейнеров.

Исследования

Выполненный обзор существующих интеллектуальных обучающих систем дает следующие виды технологий в интеллектуальных обучающих системах:

- построение последовательности курса обучения;
- интеллектуальный анализ ответов обучаемого;
- интерактивная поддержка в решении задач;
- помощь в решении задач, основанная на примерах.

Первым этапом разработки является классификация по типам основным структурам данных контейнерных классов [2].

Все контейнерные классы языка Java можно разделить на два типа: последовательные и ассоциативные. В C++ добавляется такой тип, как контейнеры – адаптеры.

Последовательные контейнеры обеспечивают хранение конечного количества однотипных величин в виде непрерывной последовательности. К ним относятся векторы (vector), двусторонние очереди (deque) и списки (list), а также так называемые адаптеры, то есть варианты, контейнеров – стеки (stack), очереди (queue) и очереди с приоритетами (priority_queue).

Ассоциативные контейнеры хранят в себе данные вида «ключ – значение» и обеспечивают быстрый доступ к ним по ключу. Эти контейнеры построены на основе сбалансированных деревьев. Существует пять типов ассоциативных контейнеров: словари (map), словари с дубликатами (multimap), множества (set), множества с дубликатами (multiset) и битовые множества (bitset).

Ядро стандартной библиотеки шаблонов включает три основных элемента: контейнеры, алгоритмы и итераторы. Они работают совместно один с другим, представляя тем самым готовые решения различных задач програм-мирования.

Контейнеры – это объекты, которые содержат другие объекты.

Каждый вид контейнера обеспечивает свой набор действий над данными. Выбор вида контейнера зависит от того, что требуется делать с данными в программе.

Алгоритмы обрабатывают содержимое контейнеров. Их возможности включают средства инициализации, сортировки, поиска и преобразования содержимого контейнеров. Многие алгоритмы работают с заданным диапазоном элементов контейнера.

Итераторы – это объекты, которые в той или иной степени действуют подобно указателям. Они позволяют циклически опрашивать содержимое контейнера практически так же, как это делается с помощью указателя при

циклическом опросе элементов массива.

Проектирование программного обеспечения обучающей системы

Для построения курса обучения главное меню состоит из иерархической последовательности вызова форм. В формах, в свою очередь, размещены элементы интерфейса, взаимодействие с которыми вызывает другие элементы или формы. На рисунке 1 показана общая структура вызовов элементов программы и их взаимодействия.

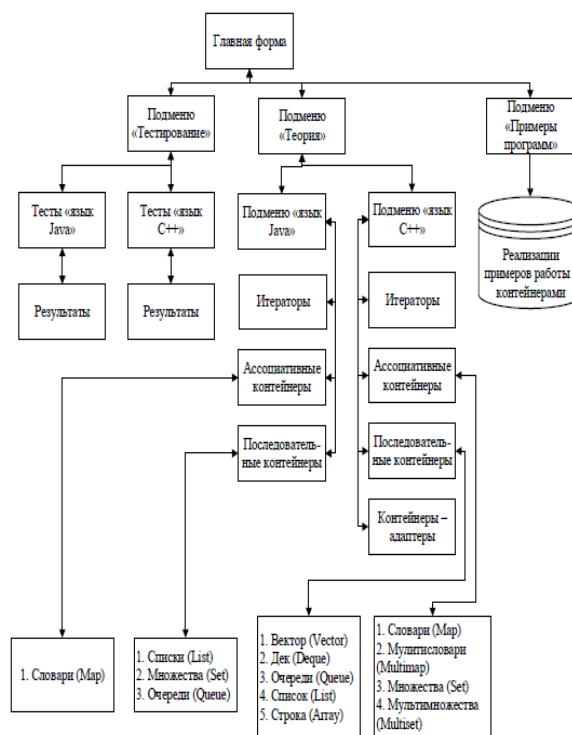


Рисунок 1 – Общая структура вызовов элементов программы и их взаимодействия

В главном меню можно выбрать один из трёх вариантов: «Теория», «Тестирование» или «Примеры программ».

В системе рассматриваются два языка программирования, для которого будет отображена теоретическая информация в соответствующих формах. Каждый язык содержит в себе теорию по различным типам контейнеров, приводятся графические схемы организации данных, как обеспечить доступ к элементам и за какое время, как будет организованно выделение памяти под очередной элемент или перераспределение памяти при удалении.

Подменю «Тестирование» позволяет генерировать случайных образов разные тесты из Базы Вопросов. Выбрав необходимый язык, открывается окно с тестовыми заданиями. Если в результате будут допущены ошибки, программа

автоматически укажет на них.

На вершине иерархии в Java Collection Framework располагаются 2 интерфейса: Collection и Map (см. рис.2). Эти интерфейсы разделяют все коллекции, входящие во фреймворк на две части по типу хранения данных: простые последовательные наборы элементов и наборы пар «ключ – значение» (словари) [3].

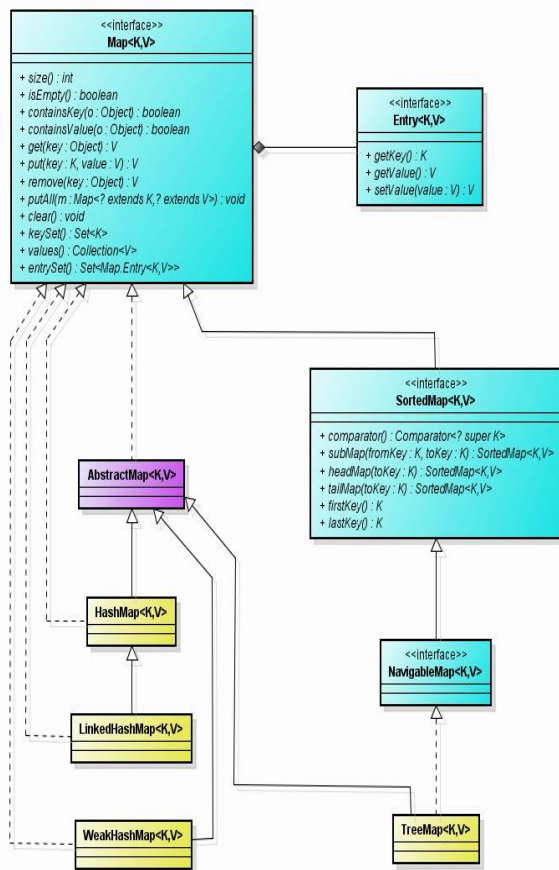


Рисунок 2 – Диаграмма классов ассоциативных контейнеров Collection Framework

Подмену «Примеры программ» содержит в себе базу программных реализаций примеров работы с контейнерными классами. Выбрав необходимый контейнер, будет запущено приложение, в котором демонстрируются функции, обрабатывающие контейнер, что позволяет изучить не только теоретическую информация по этому виду объекта, но и обеспечить выбор набора действий над данными (программную реализацию).

Теоретическая информация по типу контейнера. Например: вектор – это структура, эффективно реализующая произвольный доступ к элементам, добавление в конец и удаление из конца; двусторонняя очередь эффективно реализует произвольный доступ к элементам, добавление в оба конца и удаление из обоих

концов; список эффективно реализует вставку и удаление элементов в произвольное место, но не имеет произвольного доступа к своим элементам и т.п.

Векторы, двусторонние очереди и списки поддерживают разные наборы операций, среди которых есть совпадающие операции. Они могут быть реализованы с разной эффективностью.

В теоретической части приводятся схемы организации данных, как обеспечить доступ к элементам и за какое время, как будет организованно выделение память под очередной элемент или перераспределение памяти при удалении.

Выбор программной реализации позволяет выбрать программу из набора, запустить ее и получить результат. Например работы с вектором: в файле находится произвольное количество целых чисел; программа считывает их в вектор и выводит на экран в том же порядке.

Реализация обучающей системы

Программная реализация создана на языке Java, т.к. он является кроссплатформенным и при помощи JavaFX и приложения Scene Builder очень просто создать необходимый пользовательский интерфейс [4].

JavaFX – платформа для создания RIA, позволяет строить унифицированные приложения с насыщенным графическим интерфейсом пользователя для непосредственного запуска из-под операционных систем, работы в браузерах и на мобильных телефонах, в том числе работающих с мультимедийным содержимым. Среди возможностей этой платформы можно отметить: кроссплатформенность, поддержка каскадных таблицей стилей, поддержка анимации компонентов, возможность работы и отображение 3D графики, поддержка сенсорных экранов и многое другое [5].

Scene Builder – приложение, при помощи которого очень легко и просто создать необходимый интерфейс. Для этого необходимо просто перетащить необходимый компонент (кнопку или текстовое поле) из библиотеки на создаваемую форму. Параметры элементов изменяются в соответственном меню. Так же можно подключить каскадные таблицы стилей (CSS), что в свою очередь позволяет создавать более насыщенный и красивый интерфейс. Приложение автоматически генерирует FXMML-код, который включается в проект.

Единственным недостатком платформы Java является обязательное наличие на компьютере пользователя виртуальной Java-машины, которая необходима для запуска приложения. Во избежание проблем с запуском приложения установочный файл с виртуальной машиной будет включён в состав архива

программы.

При нажатии на кнопку «ТЕОРИЯ» и выборе языка программирования (Java или C++) открывается окно тип контейнера последовательный или ассоциативные (см. рис.3).

Выбрав меню «Списки» открывается окно с информацией о данном интерфейсе, а также есть возможность переключения между теоретической информацией о каждом контейнере.

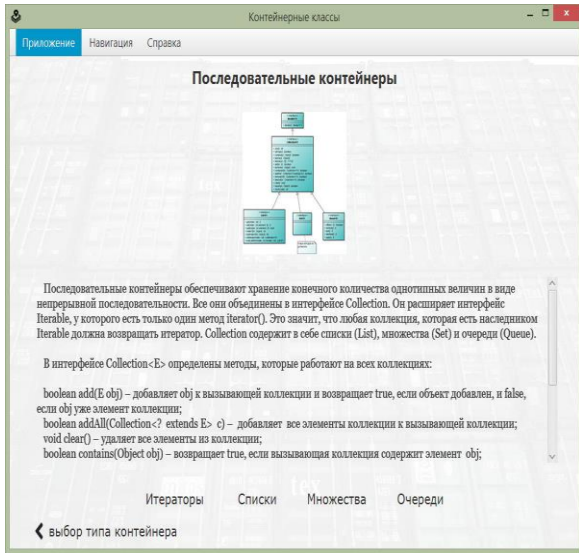


Рисунок 3 – Последовательные контейнеры

Кнопка «принцип работы...» открывает окно, в котором визуализирована и детально описана схема работы конкретного контейнера (см. рис.4).

Принцип работы меню «Множества» и «Очереди» подобен меню «Списки» и отличается лишь составом контейнеров.

Аналогично осуществляется работа при изучении ассоциативных контейнеров.

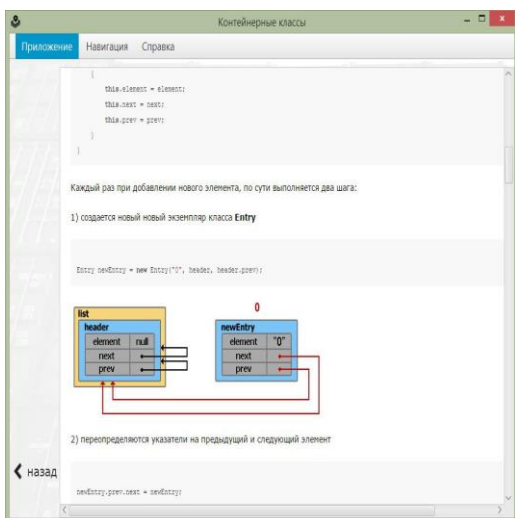


Рисунок 4 – Принцип работы контейнера

Меню «ТЕСТИРОВАНИЕ» открывает генерацию теста. Можно выбрать только один правильный ответ, но некоторые задания имеют несколько правильных ответов (см. рис.5). Допущенные ошибки выделяются красным цветом. После окончания работы пользователя система анализирует сумму ответов.

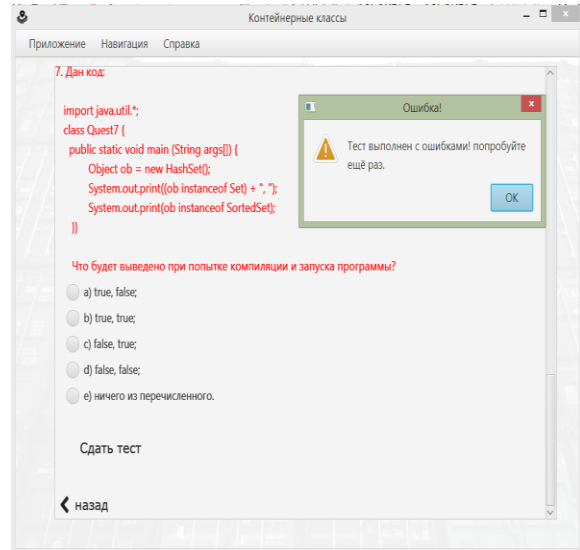


Рисунок 5 – Окно тестирования

Меню «ПРИМЕРЫ ПРОГРАММ» открывает окно со списком доступных примеров программ. При выборе одного из пунктов открывается приложение, демонстрирующее возможности контейнера (см. рис.6).

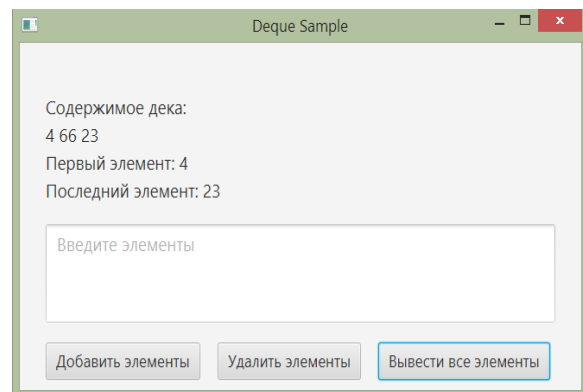


Рисунок 6 – Пример программы работы с деком

Выводы

Были изучены основные принципы проектирования обучающих систем; разработана последовательность курса обучения контейнерных классов высокоуровневых языков программирования: Java, так называемый Collections Framework, и C++ – STL контейнеры; спроектирована общая структура вызовов

элементов программы и их взаимодействия; разработан программный продукт, который позволяет изучить и эффективно использовать контейнерные классы.

Достоинство данной разработки заключается в том, что такая система дает не только теоретические знания в качестве справочника и графического представления объектов (структуры распределения и выделения памяти), но и позволяет провести интеллектуальный анализ ответов обучаемого, а также в интерактивном режиме проверить работоспособность алгоритмов работы с различными типами контейнерных классов.

Разработанную систему можно использовать в ВУЗах для обучения студентов контейнерным классам, а так же любому начинающему программисту.

Литература

1. Павловская Т. А. С/С++. Программирование на языке высокого уровня /

Павловская Т. А. – СПб.: Питер, 2003. – 461 с.

2. Ольшевский А. И. Разработка обучающего пакета программ работы с контейнерными классами / А. И. Ольшевский, А.В. Стась // Сборник материалов VII Международной научно-технической конференции в рамках II Международного Научного форума Донецкой Народной Республики 26 мая 2016г. «Информатика, управляющие системы, математическое и компьютерное моделирование» (ИУСМКМ-2016). – Донецк, С.523-525

3. Шилдт Г. С++: базовый курс / Шилдт Г. – [3-е изд.]; [пер. с англ.]. – М.: Вильямс, 2010. – 624 с.

4. Контейнеры STL. [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru/library/1fe2x6kt.aspx>

5. Коллекции в Java. [Электронный ресурс]. – Режим доступа: <http://www.quizful.net/post/Java-Collections>

Ольшевский А.И. Обучающая система работы с контейнерными классами. Рассмотрена классификация по типам основных структур данных контейнерных классов. Описаны возможности реализации их на языке Java с помощью коллекций, а в С++ – библиотекой STL. Определен подход формирования поддерживающий разные наборы операций, среди которых есть совпадающие операции. Реализован простой и понятный в использовании пользовательский интерфейс с подсказками и справкой по теоретической части для каждого типа контейнерных классов, а также примерами программ для конкретной реализации. Проанализирована эффективность разных методов соответствующих операций.

Ключевые слова: структуры данных, контейнерные классы, наборы операций, эффективность разных методов.

Olshevsky AI Training system for working with container classes. Classification by types of main data structures of container classes is considered. The possibilities of implementing them in the Java language with the help of collections are described, and in C++ the library of STL. A formation approach has been defined that supports different sets of operations, among which there are coincident operations. Implemented a simple and intuitive user interface with tips and help on the theoretical part for each type of container classes, as well as examples of programs for a particular implementation. The effectiveness of different methods of corresponding operations is analyzed.

Keywords: data structures, container classes, operation sets, efficiency of different methods.

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

Разработка функциональной структуры логистической системы формирования заказов для интернет-магазина

В.А. Светличная, Н.К. Андриевская, К.Ю. Чаленко
Донецкий национальный технический университет
prostowebiste@gmail.com

Светличная В.А., Андриевская Н.К., Чаленко К.Ю. Разработка функциональной структуры логистической системы формирования заказов для интернета-магазина. В статье рассматриваются основные типы и задачи логистики, обосновывается необходимость разработки логистической системы интернет-магазина. Проводится обзор аналитических инструментов сбора статистики и описываются основные критерии выбора закупаемой продукции и возникающие при этом ограничения. При разработке моделей логистической системы формирования заказов для интернет-магазина реализованы диаграмма прецедентов и диаграмма кооперации. Также были выделены основные пользователи информационной системы, разработана их иерархия и описаны внешние компьютеризированные информационные подсистемы, взаимодействующие с логистической системой. Приведенные модели будут использованы при дальнейшей разработке обобщенной объектной модели и других детальных диаграмм UML для логистической системы формирования заказов интернет-магазина с учетом оптимальных вариантов закупок товара.

Ключевые слова: функциональная структура, модель, логистика, заказ, интернет-магазин, товар, затраты, UML, диаграмма использования, диаграмма кооперации

Введение

Сегодня все больше продаж приходится на интернет-магазины, которые стремительно развиваются и пользуются популярностью. Online-покупки привлекают потребителей возможностью сэкономить финансы и время, а также получить товар, не выходя из дома. Однако для магазинов специфика интернет-продаж требует больших расходов на доставку товара, оптимизация которой — одно из основных условий повышения маржинальности интернет-магазина[1].

Широко известно, что логистика определяет принципы управления организацией совместной деятельности всех функциональных подразделений предприятия по прохождению товарных потоков от поставщиков сырья через производственное предприятие к конечным потребителям.

Логистика в онлайн-коммерции имеет свои особенности и ее можно разделить на 3 основных блока:

- движение продукции от поставщика, задачами которого занимается закупочная логистика;
- хранение товара на складе - зона ответственности складской логистики;
- перемещение продукции со склада к конечному покупателю - зона

ответственности специалистов по транспортной логистике[2].

Логистические процессы Интернет-торговли существенно отличаются от торговли офлайн — скорости процессов значительно выше, программное обеспечение зачастую уникально и разрабатывается под особенности каждого интернет-магазина, а логистическая цепочка строго упорядочена. Сбой на любом этапе ведет к потере лояльности клиентов, а значит потере прибыли.

Правильно выстроенная логистика магазина позволяет получить конечными потребителями высокого качества обслуживания и доставки. Она является основой построения долгосрочных доверительных взаимоотношений между продавцом и покупателем. Как следствие, возникает необходимость организовать логистику интернет-магазина таким образом, чтобы с одной стороны магазин приносил максимально возможную прибыль, а с другой стороны клиенты оставались довольны во всех смыслах.

Возникает вопрос, при каком объеме продаж имеет смысл организовывать отдельную службу логистики. Согласно мнению экспертов интернет-магазина, у которого масштаб отправок заказов в месяц составляет от 1 до 500, нет необходимости выстраивать логистические процессы — подготовку отправок можно делегировать одному из сотрудников компании. Если же месячный объем

отправлений достигает от 500 до 3 тысяч, тогда “фулфилмент” (англ. fulfilment — выполнение) выгодней организовать самостоятельно, поручив задачи отделу продаж. “Фулфилмент” - это комплексная услуга, при которой логистический оператор берет на себя прием товаров от поставщика, хранение, комплектацию, упаковку, организацию доставки, прием денег покупателей и обработку возвратов. Благодаря такому аутсорсингу интернет-магазин может полностью сконцентрироваться на маркетинге и продажах[2].

Таким образом, в случае, когда рассматриваемый интернет-магазин существует без корректно организованной логистической системы, но при его функционировании существует возможность собирать статистику продаж при помощи аналитических инструментов, и у магазина значительно больше 500 отправок в месяц, а также учитывая мнение экспертов, можно сделать вывод о необходимости организации собственного логистического центра.

Основная цель статьи - определить направления развития логистики в конкретно существующем интернет-магазине, выявить факторы, влияющие на различных этапах логистических процессов, проанализировать имеющуюся модель функционирования магазина, разработать новую функциональную модель с учетом логистики, выявить ограничения.

Постановка задачи исследования

Для реализации логистической системы необходимо проанализировать существующие процессы функционирования интернет-магазина, включающие как запросы, продажи, отказы, так и обеспечение интернет-магазина товаром. Результаты такого анализа позволят спроектировать корректную логистическую систему, которая позволит прогнозировать и закупать товар в требуемых количествах в зависимости от спроса, учитывая специфику поставщиков и предприятия. Также позволят определить основные критерии выбора закупаемой продукции, а в дальнейшем разработать функциональную модель подсистемы, в том числе и показать взаимодействие логистической подсистемы с существующей ИС типового интернет-магазина.

Результаты исследований

Главная задача логистики заключается в минимизации затрат, которые включены в себестоимость конечного продукта, и если говорить об интернет-магазине, то задача сводится к уменьшению затрат на покупку, доставку и хранение товара.

Исходя из составляющих полного логистического процесса, выделяют три основных

вида логистики: закупочная, складская и транспортная[2].

Рассмотрим закупочную логистику. Ее предметом является собственно покупка необходимого продукта, и основными целями в данном случае выступают максимальная экономическая выгода, минимальные затраты трудовых ресурсов, а также оптимальные сроки, в которые укладывается закупка. Соответственно, задачи, которые охватывает закупочная логистика, выглядят следующим образом:

- определение потребностей покупателя;
- ранжирование основных критериев выбора закупаемой продукции;
- анализ рынка;
- выбор поставщика;
- создание и ведение баз данных;
- проведение операций по закупке;
- контроль над своевременной поставкой товара.

Для успешной деятельности интернет-магазина качественная организация закупочной логистики крайне важна.

Складская логистика заключается в оптимальной организации полного цикла складских работ, а именно: приема, обработки, хранения и отгрузки продукции со складов.

В качестве основных задач выделяют:

- перемещение товаров;
- инвентаризация продукции;
- списание, а также оприходование материальных средств;
- расчет себестоимости продукции;
- учет остатков товара на складе;
- обороты за отчетный период и др.

Ключевыми являются отлаженные процессы в службе доставки и на складе. Если говорить о показателях эффективности, которые могут отслеживаться, это:

- процент затрат на логистику в цене товара;
- количество заказов, доставленных в срок по отношению к общему числу заказов;
- количество заказов, сформированных и доставленных полностью по отношению к общему числу заказов;
- количество заказов, доставленных без нарушения упаковки к общему числу заказов.

Транспортная логистика определяется как грамотная организация процесса перемещения товара из одной точки в другую при минимальных сроках, трудозатратах и материальном обеспечении. Перед ней стоят такие задачи:

- выбор подходящего транспорта;
- выбор способа транспортировки;
- выбор перевозчика и других логистических партнеров;

- определение рациональных маршрутов доставки;
- обеспечение технологического единства транспортно-складского процесса;
- оптимизация параметров транспортного процесса (увеличение скорости транспортировки, уменьшение расходов топлива и пр.).

Согласно социологическим исследованиям, покупатели через интернет среди главных критериев для выбора интернет-магазина выделяют наличие необходимого товара и своевременную, качественную доставку заказов. В оценке качества транспортной логистики покупателем «виртуального прилавка» существенное значение имеет эмоциональное восприятие: вежливость работников доставки, внимательное и заботливое отношение к клиенту, возможность отслеживать состояние груза в процессе его транспортировки и т.д.[3].

Своевременная и быстрая доставка товара покупателю наряду с качеством самого товара является краеугольным камнем успешной работы и дальнейшего продвижения в интернет-бизнесе[4].

Если проанализировать задачи различных видов логистических систем, то можно сделать вывод, что главная задача логистики заключается в минимизации затрат, которые включены в себестоимость конечного продукта[5].

Перед созданием новой системы логистики были проанализированы готовые компоненты, которые уже внедрены в онлайн-

магазине.

Опишем функциональную структуру существующей информационной системы (ИС) с помощью UML (Unified Modeling Language — унифицированного языка моделирования), языка графического описания для объектного моделирования систем[6].

Для описания функциональности ИС и взаимодействия с пользователями и внешними информационными системами в нотации UML на практике разработчиками используются диаграммы вариантов использования Use-Case, при этом каждый вариант использования описывает видимую пользователем функцию[7-9].

Функциональная структура типового интернет-магазина без использования системы логистики в виде диаграммы вариантов использования приведена на рисунке 1.

Основная работа по управлению контентом и заказами возложена на оператора сайта. Функциональные модули интернет-магазина активно взаимодействуют с ИС склада.

В таком случае в полной мере организована только складская логистика. Присутствует складское помещение, которое вмещает определенное количество продукции на продажу. Свободное место освобождается по мере продажи. За инвентаризацию отвечают складские работники. Списание, оприходование, расчет себестоимости, анализ и учет остатков товара осуществляет программа складского учета, которая связана с интернет-магазином.

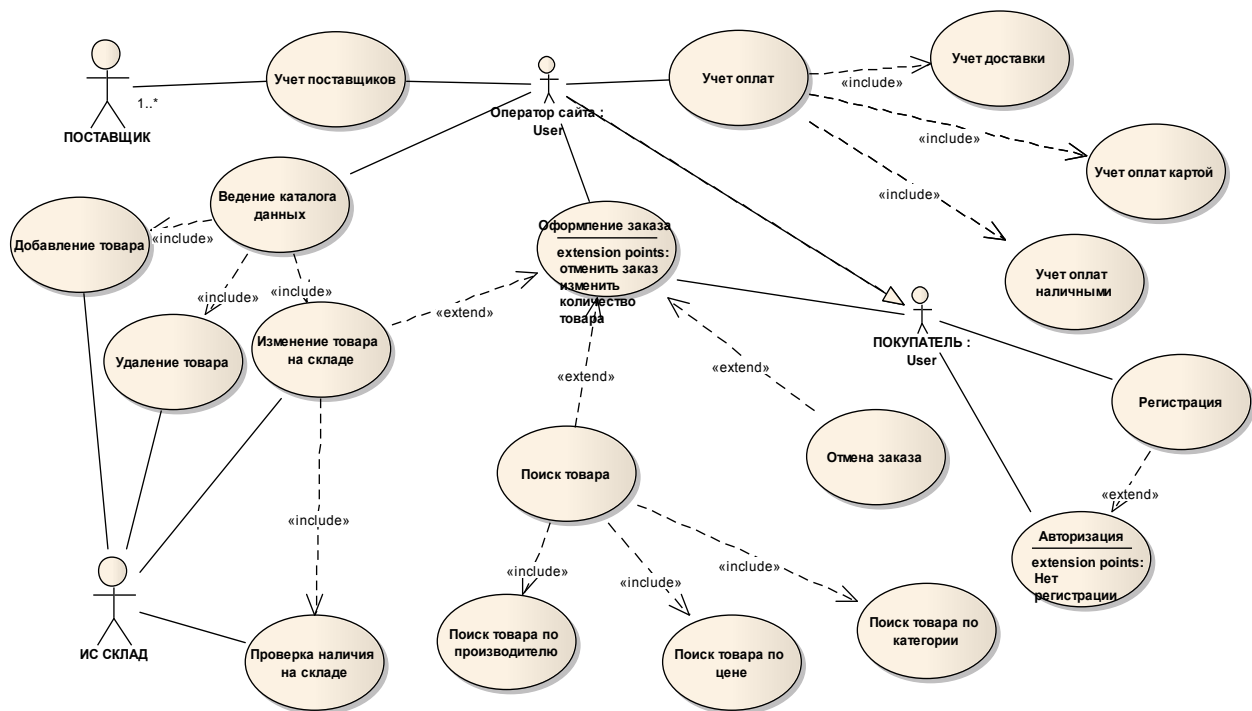


Рисунок 1 - Функциональная структура интернет-магазина без использования системы логистики

Что же касается закупочной логистики, то существует только список поставщиков-фабрик, их ассортимент, возможности поставок, сроки выпуска продукции и ее стоимость. При этом прайс-листы необходимо заново запрашивать перед каждой закупкой, т.к. стоимость отдельных позиций может меняться.

Каждая фабрика производит по меньшей мере одну из пяти основных категорий товара магазина. В нижеприведенной таблице представлены соответствующие данные по категориям товара и предоставляющим их предприятиям (см. табл. 1).

Таблица 1. Сопоставление категорий товара с производящими их фабриками

| Категория 1 | Категория 2 | Категория 3 | Категория 4 | Категория 5 |
|-------------|-------------|-------------|-------------|-------------|
| Фабрика 1 | Фабрика 2 | Фабрика 3 | Фабрика 4 | Фабрика 5 |
| Фабрика 6 | Фабрика 6 | Фабрика 7 | Фабрика 7 | Фабрика 8 |
| Фабрика 9 | Фабрика 10 | Фабрика 10 | Фабрика 9 | Фабрика 10 |
| Фабрика 10 | | | Фабрика 10 | |

Фабрики 1-5 – это основные поставщики, которые поставляют исключительно большие партии продукции по низкой цене, и отличаются также довольно долгой доставкой.

Фабрики 6-9 – имеют возможность поставлять продукцию нескольких категорий, к тому же менее масштабными партиями, чем Фабрики 1-5. Они также отличаются более быстрой доставкой, однако и стоимость работы с ними больше, чем при первом варианте. Фабрика 10 может доставить любой товар в любых количествах за короткие сроки, но себестоимость такого товара будет сильно завышена.

Рассмотрим основные моменты, на которые следует обратить внимание при разработке логистической системы.

Во-первых, это сбор статистики и ведение аналитического учета. Определение потребностей клиентов является основой при выборе закупаемого товара. Для составления реалистичного списка потребностей необходимо использовать веб-аналитику работающего онлайн-магазина. Аналитика представляет собой непрерывный сбор, анализ и обработку сведений о действиях клиентов.

В разрабатываемой логистической системе интернет-магазина есть возможность автоматически определять востребованность каждой позиции товара. Сайт будет связан с двумя основными аналитическими системами – Google Analytics и Яндекс.Метрика, которые позволяют определить количество покупок, конверсию и, конечно же, популярность конкретных товаров или продуктов. Эти параметры дают возможность спрогнозировать время, когда необходимо пополнить запасы определенного товара, а также помогают определиться с объемами закупок.

Во-вторых, необходимо определить, когда и какой товар нужно закупать. На складе хранится приблизительно 3000 позиций. К сожалению, докупать единицы товара по мере их покупки потребителями – решение нерентабельное из-за издержек при доставке

заказа на склад. Поэтому важно понимать, когда нужно провести закупку. Таким образом, для каждого товара, зная среднее время его реализации, можно вычислить порог количества, при котором необходимо делать заказ (см.рис.2).



Рисунок 2 – График наличия товара на складе

При этом порог для нового заказа \min высчитывается по следующей формуле:

$$\min = t_{d \max} * \bar{t}_s,$$

где $t_{d \max}$ – максимальное время доставки товара, дней;

\bar{t}_s – среднее время покупки товара покупателями, шт./день.

В момент, когда количество позиций какого-либо товара опускается до числа “min”, необходимо проводить закупку. Время между “min” и “null” – максимальное время, за которое возможно доставить новую партию товара на склад.

Например, мы знаем, что максимальное количество товара «А» на складе – 1000 ед. По имеющейся статистике можно понять, что каждый день через интернет-магазин покупается в среднем 3 единицы данного товара. Самый выгодный по цене вариант доставки занимает 2-3 месяца по при минимальном размере партии в 800 ед. Из этого следует, что критический порог, при котором необходимо проводить закупку:

$$\min = (3 \text{ мес.} * 30 \text{ дней}) * 3 \text{ ед./день} = 270 \text{ ед.}$$

Округляем до сотен в качестве страховки на случай непредвиденной задержки, и получаем, что при самых худших условиях времени

доставки нам необходимо делать закупку при падении количества товара «А» до порога в 300 единиц. Далее данный товар заносится в базу данных как нуждающийся в закупке. По достижению определенного количества позиций, которые необходимо заказать, формируется запрос для подходящей фабрики.

И наконец, если есть возможность определить, когда и какой товар нужно закупать, то появляется вопрос – какие же основные критерии выбора закупаемой продукции? Важным критерием, который можно вычислить при помощи аналитики является рентабельность совершённых закупок и магазина в целом.

В случае с интернет-магазином существует несколько ограничений.

Это размер складского помещения, стоимость товара, количество оборотных средств и поставщики. Если рассматривать эти пункты отдельно, то получается, что размер склада не составит больших проблем, если вычислить порог каждого товара и производить заказы только при достижения этого самого порога. Также стоит учитывать, что складская логистика уже успешно работает на предприятии. Стоимость товара будет варьироваться в зависимости от типа доставки.

Поставщики и доставка же требуют более тщательного рассмотрения. После выбора подходящей фабрики с ней необходимо связаться. Далее вести переговоры, в которых есть много важных нюансов, кроме окончательной цены. Требуется обязательно обсудить качество товара, условия оплаты, сроки поставок и ваши дальнейшие отношения с поставщиком продукции. Кроме того, у любой компании есть нижняя планка цены – минимальное рентабельное значение. Выгоднее всего привозить легкий по весу товар, который можно заказывать большими объемами. Так как товар магазина достаточно легкий, можно выделить 4 наиболее подходящих способа доставки[10].

Курьерскую доставку можно использовать, если товар уже оплачен покупателем и можно быть уверенным, что нет никаких рисков. Среди подобных служб можно выделить DHL, FedEx и EMS. Стоимость вычисляется из расчета за вес. Работая с курьерскими службами, можно не беспокоиться о решении бумажных и таможенных вопросов[10]. Эти службы позволяют получать товары быстро, но стоимость редко оправдывает этот способ. Международную почту обычно используют для доставки образцов товара, правда по срокам это занимает до пары месяцев. Нет возможности отследить местонахождение посылки и есть вероятность, что посылку могут потерять.

Третий вариант - воздушный фрахт. Обычно доставка занимает 5-10 дней, стоимость ее ниже, чем в вышеуказанных способах, однако придется самостоятельно решать все таможенные

вопросы.

Самый популярный способ доставки – морской фрахт. Он требует довольно много времени. К тому же, придется самостоятельно разбираться с бумагами, или воспользоваться услугами экспедирующей компании. Стоимость конечного продукта будет минимальная, но заказ придется делать за несколько месяцев. Также необходимо заключить договор с экспедирующей компанией/компаниями для экономии времени прохождения таможи.

Таким образом, логистическая система интернет-магазина должна заниматься решением следующего ряда вопросов:

- организация качественного аналитического аппарата для мониторинга ситуации на рынке – как с потребителями, так и с поставщиками;
- построение рациональной системы товарооборота для точного определения объёмов закупок и их сроков;
- составление наиболее оптимального списка поставщиков;
- подбор подходящих средств и методов доставки.

Следует отметить, что при подходе к вышеперечисленным задачам необходимо учитывать ограничения, продиктованные индивидуальными особенностями предприятия, для которого разрабатывается система, а именно:

- объем имеющихся складских помещений и возможность оперативного расширения рабочей площади в случае непредвиденных обстоятельств;
- запас доступных для использования финансовых средств и наличие кризисного фонда;
- необходимость учёта особенностей работы различных поставщиков (ценовая политика компании и её стабильность, разнообразие и стоимость доставки товара).

Разработанная UML диаграмма логистической системы интернет-магазина, описывающая функциональную структуру, приведена на рисунке 3.

Как видно из диаграммы, структура такой системы гораздо более сложная, чем структура типового интернет-магазина. Также произошло перераспределение части функций ИС интернет-магазина между новыми сотрудниками в связи с выделением отделов продаж, закупок, логистики в отдельные структурные единицы. Немаловажным моментом является и уменьшение нагрузки на оператора сайта в условиях постоянно возрастающего количества продаж и популярности интернет-магазина, ИС которого взаимодействует с логистической подсистемой в качестве внешней информационной системы (см. рис.4).

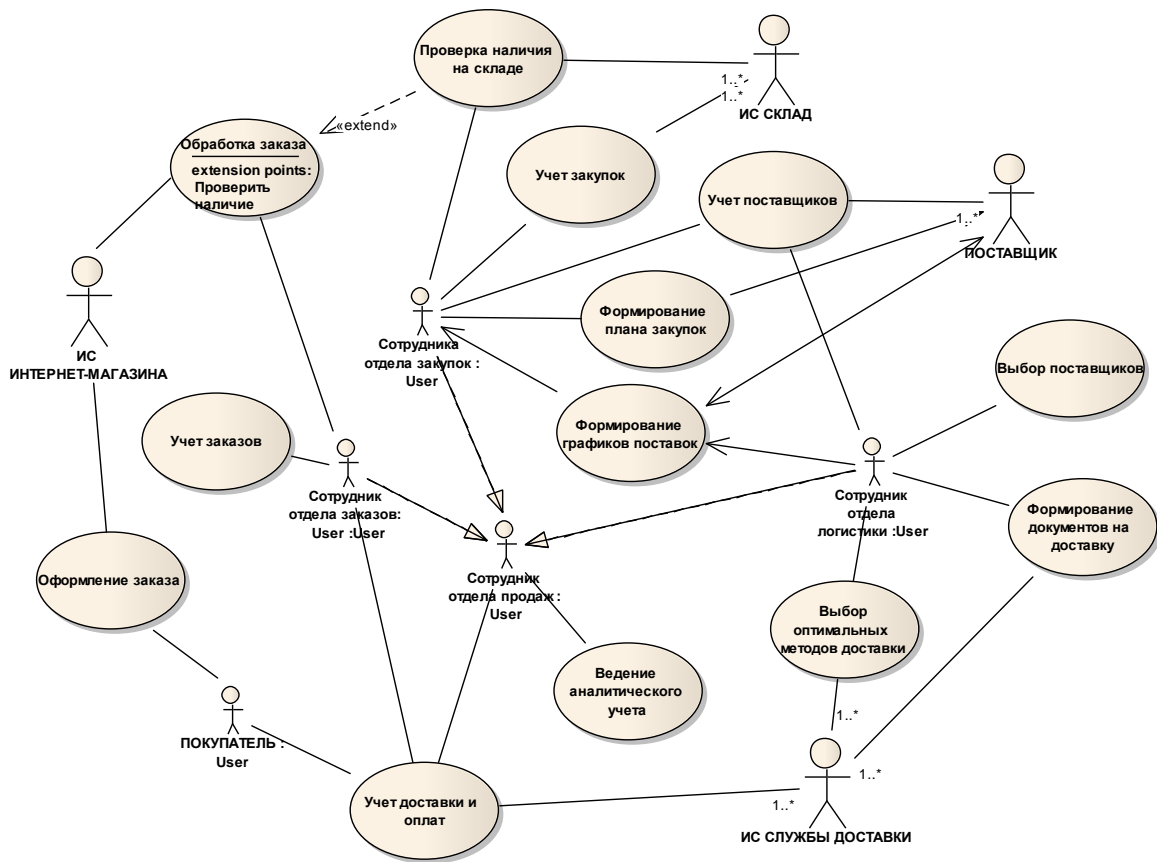


Рисунок 3 – Функциональная структура логистической системы интернет-магазина в виде Use-case диаграммы

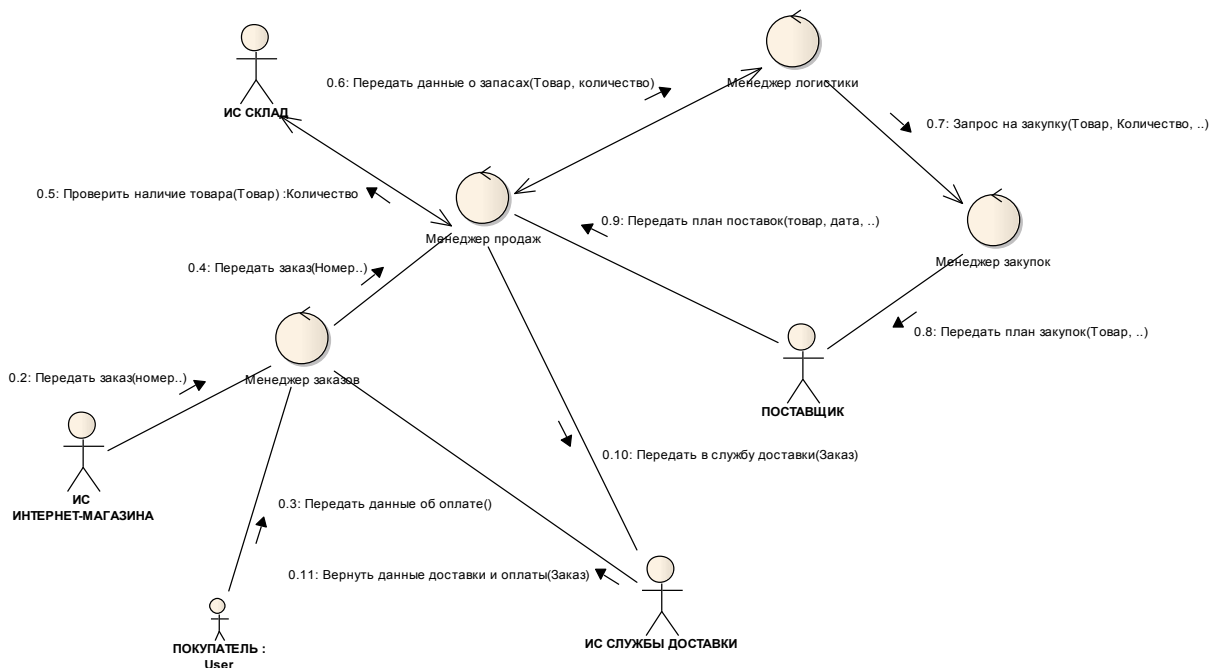


Рисунок 4 – Диаграмма взаимодействия основных объектов логистической системы формирования заказов для интернет-магазина

Процесс взаимодействия разрабатываемой логистической системы формирования заказов с ИС интернет-магазинов предполагается реализовать через информационное и программное обеспечение с помощью личного кабинета, возможности которого регулярно расширяются и совершенствуются. Здесь клиент может оформлять заявки и контролировать состояние своих заказов. Также логистическая система для сайта даст возможность получать финансовые отчеты каждую неделю.

Что касается взаимодействия с клиентами интернет-магазинов и других сервисов, таких как ИС служб доставки или поставщика, будет осуществляться оповещение о времени и дате доставки, названии предприятия-отправителя с именем телефона и именем курьера с помощью смс-оповещения, выполняться обмен данными о наличии товара на складе при обработке заказов.

Для отображения взаимодействия основных объектов логистической системы формирования заказов для интернет-магазина была выбрана диаграмма кооперации, которая показывает структурную организацию обменивающихся сообщениями основных объектов логистической системы (см. рис.4).

В результате проведенных исследований о необходимости разработки логистической системы интернет-магазина выделены задачи такой системы и ограничения, накладываемые на принятия окончательных решений. Разработка собственного логистического центра для интернет-магазина уменьшит затраты, которые включены в себестоимость конечного продукта и поможет автоматизировать отдельные элементы работы компании и увеличить товарооборот.

Выводы

Проведен анализ имеющейся модели работы компании и интернет-магазина. Выполнен обзор основных видов логистики. Рассмотрены способы прогнозирования закупок товара. Обоснована необходимость организации собственного логистического центра. С помощью средств языка моделирования UML разработана диаграмма Use-case, отображающая функциональную структуру будущей логистической системы, представлена укрупненная UML-диаграмма взаимодействия объектов разрабатываемой логистической системы. Следующим этапом работы должна стать детальная разработка логистической системы с выбором и обоснованием методов определения наиболее оптимальных вариантов закупок товара и разработкой детальных диаграмм последовательности, деятельности и построения обобщенной объектной модели логистической системы формирования заказов

для интернет-магазина.

Литература

1. Какая логистика нужна интернет-магазину. [Электронный ресурс] – Режим доступа: <https://trademaster.ua/logistic/1706> (Дата обращения: 08.05.2017).
2. Как организовать логистику в интернет-магазине [Электронный ресурс] – Режим доступа: <https://www.shopolog.ru/metodichka/logistics/kak-organizovat-logistiku-v-internet-magazine/> (Дата обращения: 08.05.2017).
3. Организация логистики для интернет-магазинов. [Электронный ресурс] – Режим доступа: <https://info.paymaster.ru/glossary/logistika-dlja-internet-magazinov/> (Дата обращения: 10.05.2017).
4. Оптимизация издержек на логистику для интернет-магазина [Электронный ресурс] – Режим доступа: <http://www.logistics.ru/eshops/news/optimizaciya-izderzhkek-na-logistiku-dlya-internet-magazina> (Дата обращения: 10.05.2017).
5. Простая логистика для интернет-магазина [Электронный ресурс] – Режим доступа: <https://ain.ua/2015/06/23/prostaya-logistika-dlya-internet-magazina> (Дата обращения: 11.09.2017).
6. Фаулер М. UML. Основы, 3-е издание. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 192 с.
7. А.И. Дорош, В.А. Светличная, Н.К. Андриевская. Функциональная структура подсистемы поддержки принятия решения для задач финансовой реструктуризации предприятия. [Электронный ресурс] – Режим доступа: <http://ea.donntu.org/handle/123456789/15391> (Дата обращения: 11.09.2017).
8. Светличная В.А., Шумаева Е.А., Андриевская Н.К. Разработка моделей для системы поддержки принятия решений при финансовой реструктуризации промышленного предприятия [Электронный ресурс] – Режим доступа: <http://ea.donntu.org/handle/123456789/29469> (Дата обращения: 11.09.2017).
9. А.И. Дорош, А.С. Тельнюк, В.А. Светличная, Е.А. Шумаева, Н.К. Андриевская. Использование нейронных сетей для определения взаимозависимости основных факторов при финансовой реструктуризации. [Электронный ресурс] – Режим доступа: <http://ea.donntu.org/handle/123456789/30165> (Дата обращения: 11.09.2017).
10. Четыре способа доставки в Россию [Электронный ресурс] – Режим доступа: <http://emagnat.ru/organizaciya-postavkok-iz-kitaya.html> (Дата обращения: 11.05.2017).

Светличная В.А., Андриевская Н.К., Чаленко К.Ю. Разработка функциональной структуры логистической системы формирования заказов для интернета - магазина. В статье рассматриваются основные типы и задачи логистики, обосновывается необходимость разработки логистической системы интернет-магазина. Проводится обзор аналитических инструментов сбора статистики и описываются основные критерии выбора закупаемой продукции и возникающие при этом ограничения. При разработке моделей логистической системы формирования заказов для интернет-магазина реализованы диаграмма прецедентов и диаграмма кооперации. Также были выделены основные пользователи информационной системы, разработана их иерархия и описаны внешние компьютеризированные информационные подсистемы, взаимодействующие с логистической системой. Приведенные модели будут использованы при дальнейшей разработке обобщенной объектной модели и других детальных диаграмм UML для логистической системы формирования заказов интернет-магазина с учетом оптимальных вариантов закупок товара

Ключевые слова: функциональная структура, модель, логистика, заказ, интернет-магазин, товар, затраты, UML, диаграмма использования, диаграмма кооперации.

Svetlichnaya V.A., Andrievskaya N.K., Chalenko K.Y. Development of the functional structure of the logistics system of formation of orders for the online store. The article discusses the main types and tasks of logistics, necessity of development of logistic system of the online store. Provides an overview of analytic tools statistics and describes the main criteria for the selection of purchased products and resulting limitations. When modelling a logistic system of formation of orders for the online shop implemented use-case diagram and cooperation diagram. Also identified the key users of the information system, developed hierarchy and describes the external computerized information subsystem that interacts with the logistics system. The models will be used in the further development of a generalized object model, and other detailed UML diagrams for the logistic system of formation of orders of online store with regard to best options for the procurement of goods.

Keywords: functional structure, model, logistics, order, online store, merchandise, cost, UML, Use-Case diagram, Cooperation diagram.

Статья поступила в редакцию 20.09.2016
Рекомендована к публикации д-ром физ.-мат. наук А.С. Миненко

УДК 004.89

Разработка системы контроля и мониторинга состояния охраны труда и промышленной безопасности на угледобывающих предприятиях

И.А. Тарасова, Г.А. Резникова
Донецкий национальный технический университет
irina_tarasova@i.ua, alinakohtan@inbox.ru

Тарасова И.А., Резникова Г.А. Разработка системы контроля и мониторинга состояния охраны труда и промышленной безопасности на угледобывающих предприятиях. Выполнен анализ угледобывающих предприятий, на основании которого разработана система контроля и мониторинга состояния охраны труда и промышленной безопасности на шахтах государственного предприятия «Донецкая угольная энергетическая компания», позволяющая минимизировать время на принятие управленческих решений, что приводит к снижению уровня производственного травматизма.

Введение

Уголь – самый распространенный в мире энергетический ресурс. Тема угольной промышленности в данный момент остается актуальной. Во-первых, потому, что в топливном балансе теплоэлектростанций уголь занимает более 90 процентов. Во-вторых, металлургическое производство невозможно без коксующегося угля. Это говорит об огромной значимости отрасли для экономики нашего региона.

Шахты отличаются сложными горно-геологическими условиями, которые продолжают ухудшаться. 90 процентов ныне действующих угледобывающих предприятий работают без реконструкции и существенной модернизации свыше 30 лет. Поэтому охрана труда и промышленная безопасность являются приоритетными вопросами в работе на угледобывающих предприятиях.

Цель статьи – снижение производственного травматизма и обеспечение безопасности производственных процессов, оборудования, зданий и сооружений, за счет разработки системы контроля и мониторинга состояния охраны труда и промышленной безопасности на угледобывающих предприятиях, позволяющей минимизировать время на принятие управленческих решений.

Для достижения поставленной цели в статье решаются следующие задачи:

- анализ угледобывающих предприятий;
- разработка модели системы контроля и мониторинга состояния охраны труда и промышленной безопасности;
- проведение концептуального проектирования, которое включает в себя разработку диаграммы потоков данных, диаграммы «сущность-связь», иерархической модели данных, сетевой модели данных и

реляционной модели данных системы «КМСОТ и ПБ»;

- выполнение программной реализации, которая включает в себя обоснование выбора системы управления, проектирование пользовательского интерфейса, а также разработку схемы данных и схемы взаимодействия компонентов системы.

Анализ угледобывающих предприятий

Одной из важнейших проблем на угледобывающем предприятии является обеспечение безопасности и безвредности производства, профилактика травматизма и заболеваний, связанных с неблагоприятными условиями труда.

Для решения указанной проблемы необходимо соблюдение соответствующих правовых и технических норм в области безопасности и гигиены труда, правильная организация трудового процесса, постоянный контроль и отчетность перед вышестоящими административными и хозяйственными органами.

Анализ основан на работе наиболее крупного угледобывающего государственного предприятия «Донецкая угольная энергетическая компания» (далее ГП «ДУЭК»), в состав которого на правах обособленных подразделений входят шесть шахт, такие как:

- ш. «Трудовская»;
- ш. им. Е.Т. Абакумова;
- ш. им. А.А. Скочинского;
- ш. «Октябрьский Рудник»;
- ш. им. М.И. Калинина;
- ш. им. Челюскинцев.

Директора шахт для контроля за состоянием и соблюдением требований

нормативных и других документов по охране труда создают службу охраны труда, целью которой является обеспечение высокопроизводительного трудового процесса, предотвращение аварийности, травматизма и профзаболеваемости при выполнении производственных заданий на предприятии, в его структурных подразделениях и на рабочих местах [1].

В свою очередь отдел охраны труда, гражданской обороны и чрезвычайных ситуаций ГП «ДУЭК» является аппаратом управления, главная функция которого обеспечение и контроль на шахтах и других обособленных подразделениях выполнения законодательных актов и нормативных документов по охране труда.

Уровень производственного травматизма, степень профессиональных заболеваний, количество несчастных случаев и аварий можно снизить за счет своевременного принятия мер.

Для того, чтобы выработать решение, отделу необходимо обработать и систематизировать большое количество информации, поэтому для автоматизации процесса обработки данных и минимизации времени на принятие решения, необходимо разработать систему контроля и мониторинга состояния охраны труда на обособленных подразделениях предприятия.

Разработка моделей системы контроля и мониторинга состояния охраны труда и промышленной безопасности

Для разработки морфологической модели службы охраны труда необходимо определить структуру подразделений компании, с которыми в дальнейшем функционирует служба охраны труда.

Структура подразделений ГП «ДУЭК» представлена на рисунке 1.

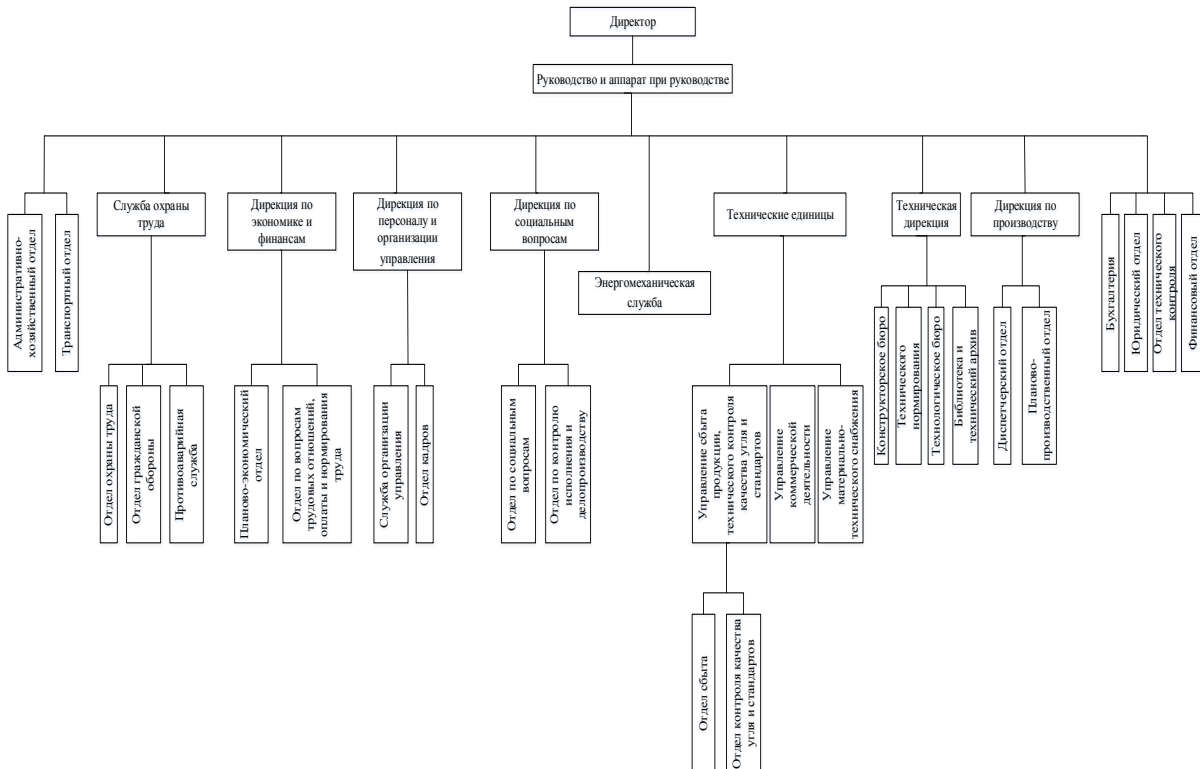


Рисунок 1 – Структура подразделений ГП «ДУЭК»

Служба охраны труда взаимодействует с другими подразделениями (отделами, цехами, службами) посредством движения нормативной документации, которая ведется на шахтах (наряд-путевки, книги нарядов, оперативный журнал и т.д.), передается в компанию для систематизации, на основании которой делаются расчеты, формируются отчеты, заявки, предложения [2].

На основании данных документов производственные участки оснащаются

горношахтным оборудованием, соответствующим горнотехническим условиям шахты, а также системами и средствами автоматизированной противоаварийной (коллективной) защиты, контрольно-измерительной аппаратурой и другими материалами.

Также на основании полученной информации разрабатываются графики проверок состояния охраны труда на участках, планы

работ для постоянно действующей комиссии по охране труда, отчеты, инструкции.

Разрабатываемая модель основывается на вербальном описании и структуре подразделений

компании. Графическая модель взаимодействия службы охраны труда с другими подразделениями ГП «ДУЭК» представлена на рисунке 2.

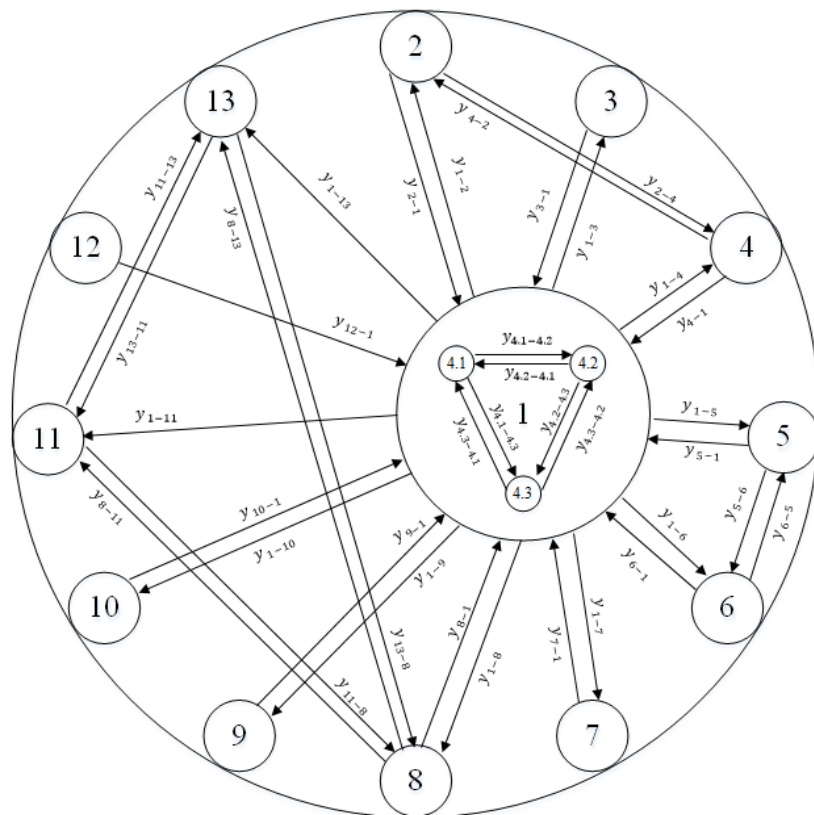


Рисунок 2 – Графическая модель взаимодействия службы охраны труда с другими подразделениями ГП «ДУЭК»

Системный анализ показал, что службе охраны труда для принятия решения необходимо обработать большое количество информации, поэтому для минимизации времени на принятие управленческих решений целесообразно провести автоматизацию работы отдела охраны труда.

На основании проведенного анализа разрабатывается модель системы «Контроль и мониторинг состояния охраны труда и промышленной безопасности» типа «черный ящик». Для этого необходимо определить элементы модели, а именно, границы системы и ее внешнюю среду, входные и выходные потоки [3]. Граница данной системы определяется совокупностью структурных подразделений ГП «ДУЭК» [4].

Внешней средой для системы являются лица ответственные за сбор, хранение и обработку поступающей информации, а также руководители, контролирующие работу, а именно:

– генеральный директор отдела охраны труда;

- специалисты отдела охраны труда;
- директор шахты;
- главный инженер;
- заместитель директора по производству;
- заместитель директора по охране труда;
- заведующий горными работами;
- заместитель главного инженера;
- главный механик (энергетик);
- главный маркшейдер (геолог);
- участковый маркшейдер (геолог);
- начальник смены;
- специалист службы охраны труда;
- горный диспетчер.

Входными данными системы являются:

- x_1 – общешахтная книга нарядов (распоряжений);
- x_2 – журнал горного диспетчера;
- x_3 – наряд-путевка горного мастера;
- x_4 – книга нарядов участка;
- x_5 – книга нарядов участка ВТБ;
- x_6 – наряд-путевка горного мастера участка ВТБ;
- x_7 – оперативный журнал горного диспетчера;

x_8 – книга нарядов службы прогноза ГДЯ;
 x_9 – наряд-путевка горного мастера службы прогноза ГДЯ;
 x_{10} – книга нарядов участка ПРТБ;
 x_{11} – наряд-путевка горного мастера участка ПРТБ;
 Выходными данными системы являются:
 y_1 – предписания;
 y_2 – график проверок состояния охраны труда на участках;
 y_3 – план работы ПДК по охране труда;

y_4 – график обучения горнорабочих и специалистов безопасным методам работы;
 y_5 – план оздоровительных мероприятий;
 y_6 – отчет экспресс-анализа травматизма, профзаболеваний и аварийности;
 y_7 – комплексная программа (статистика исправленных нарушений, количество проведенных мероприятий по данным направлениям).
 Модель системы типа «черный ящик» представлена на рисунке 3.



Рисунок 3 – Модель «черный ящик» системы «Контроль и мониторинг состояния охраны труда и промышленной безопасности»

Оперативный учет и экспресс-анализ травматизма на шахте осуществляется ежемесячно [5], ежеквартально составляется краткий отчет (динамика) по установленной Министерством угля и энергетики форме, который передается в службу охраны труда ГП «ДУЭК» для установления причин и тенденций травматизма, разработки отраслевых мероприятий для его предотвращения.

Концептуальное проектирование системы контроля и мониторинга состояния охраны труда и промышленной безопасности

Диаграмма потоков данных системы контроля и мониторинга состояния охраны труда и промышленной безопасности на угледобывающих предприятиях представлена на рисунке 4. Диаграмма отображает всех участников (пользователей) системы, а также данные, которыми обмениваются все подсистемы [7]. База данных имеет три уровня доступа. Первый – это сами шахты, которые вносят данные о проверках, проведенных руководящими лицами и специалистами шахты. Второй – это ГП «ДУЭК», который имеет доступ

к информации по всем шахтам. Статистическая информация предоставляется генеральным директором отдела охраны труда генеральному директору компании. И третий – это Министерство угля и энергетики, которое взаимодействует с подсистемой компании для доступа к информации, для дальнейших селекторных совещаний и принятия управленческих решений на основе обработанной информации.

Следующим наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD). Цель используемого моделирования данных состоит в обеспечении разработчика ИС концептуальной схемой базы данных в форме одной модели или нескольких локальных моделей, которые относительно легко могут быть отображены в любую систему баз данных [11].

С помощью диаграмм определяются важные для предприятия и его обособленных подразделений объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). Поэтому, ERD непосредственно будет использоваться для проектирования данной реляционной базы данных.

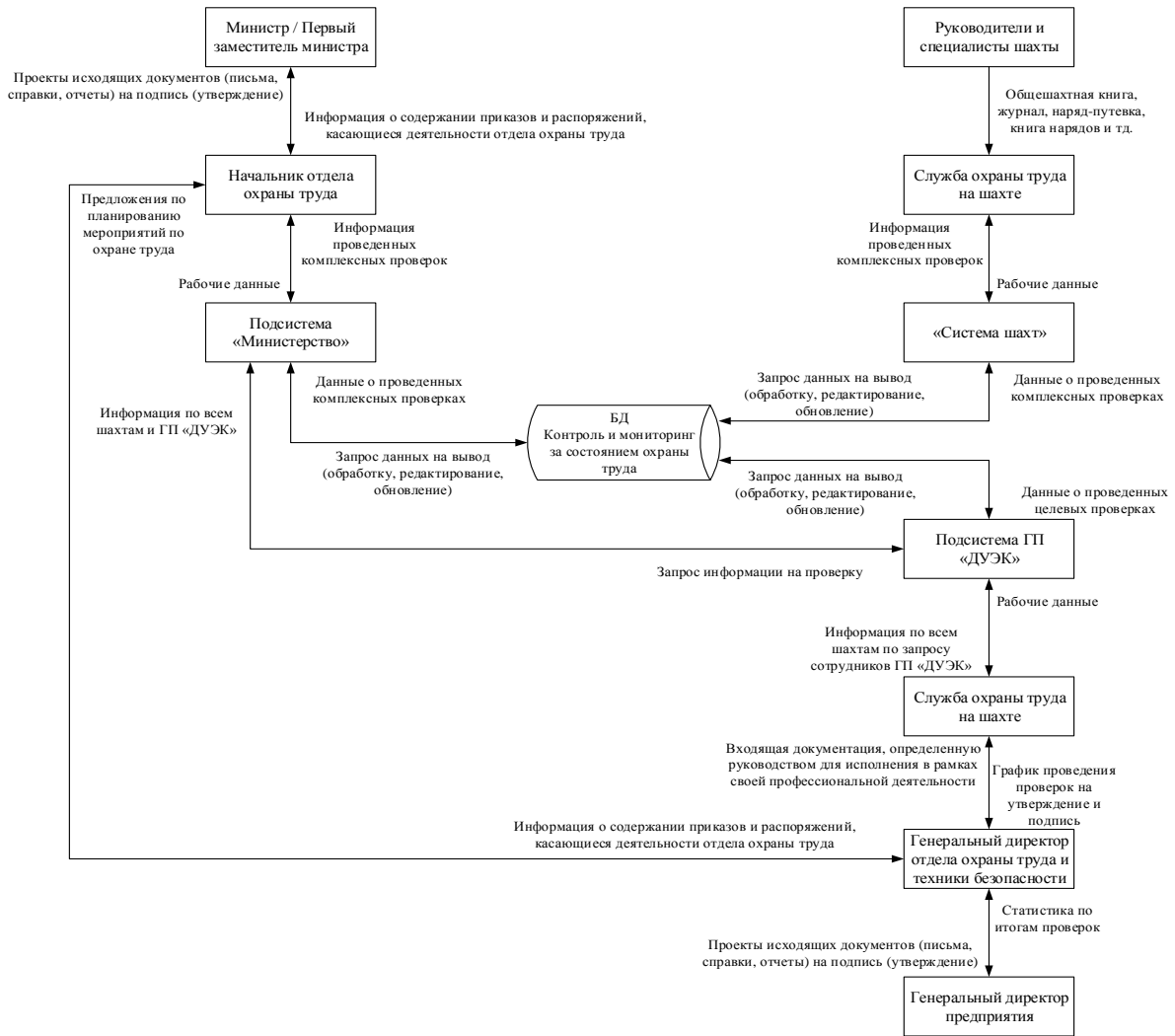


Рисунок 4 – Диаграмма потоков данных

В разрабатываемой базе данных основными объектами (сущностями) являются таблицы «Предписание (ДУЭК)», «График», «ДУЭК», «Система шахт», «Дата», «Отдел», «Вид проверки», «Сотрудники», «Должность (ДУЭК)», «Обособленные подразделения», «Смена», «Должность (шахта)», «Проверки».

Объекты (сущности) «Дата», «Отдел», «Вид проверки», «Сотрудники», «Должность (ДУЭК)», «Обособленные подразделения», «Смена», «Должность (шахта)», «Проверки» в той или иной степени выступают в качестве справочника, но идентифицируются как самостоятельные объекты. Такое разбиение информации называется приведением к третьей нормальной форме (3НФ), суть формы в том, чтобы связь между объектами была «один ко многим», что упрощает работу реляционной базы данных [11].

Диаграмма «сущность-связь» системы контроля и мониторинга состояния охраны труда и промышленной безопасности представлена на рисунке 5.

Далее устанавливается характер связей между объектами. На каждой шахте после того, как назначенная проверка завершилась, руководители и специалисты предоставляют предписание и фиксируют свое посещение в журнале. Ставится дата проверки, указывается ответственное лицо, наименование проверки и количество выявленных нарушений.

Таким образом, один объект нуждается в нескольких справочниках, которые будут хранить в себе нужную информацию, поэтому степень связи один ко многим.

«Донецкая угольная энергетическая компания» в свою очередь проводит проверки на всех обособленных подразделениях компании, вид проверки может меняться в зависимости от внешних обстоятельств, но всегда есть единая, проходящая один раз в месяц комплексная проверка. Также, после ее проведения сотрудниками отдела ОТ и ЧС предоставляется предписание, где указывается суть нарушений шахтой, устанавливаются сроки их устранения и по итогу предоставляется копия предписания по

проведенным мероприятиям. Таким образом, чтобы организовать работу одного отдела,

необходимо создать более пары объектов и не один справочник.

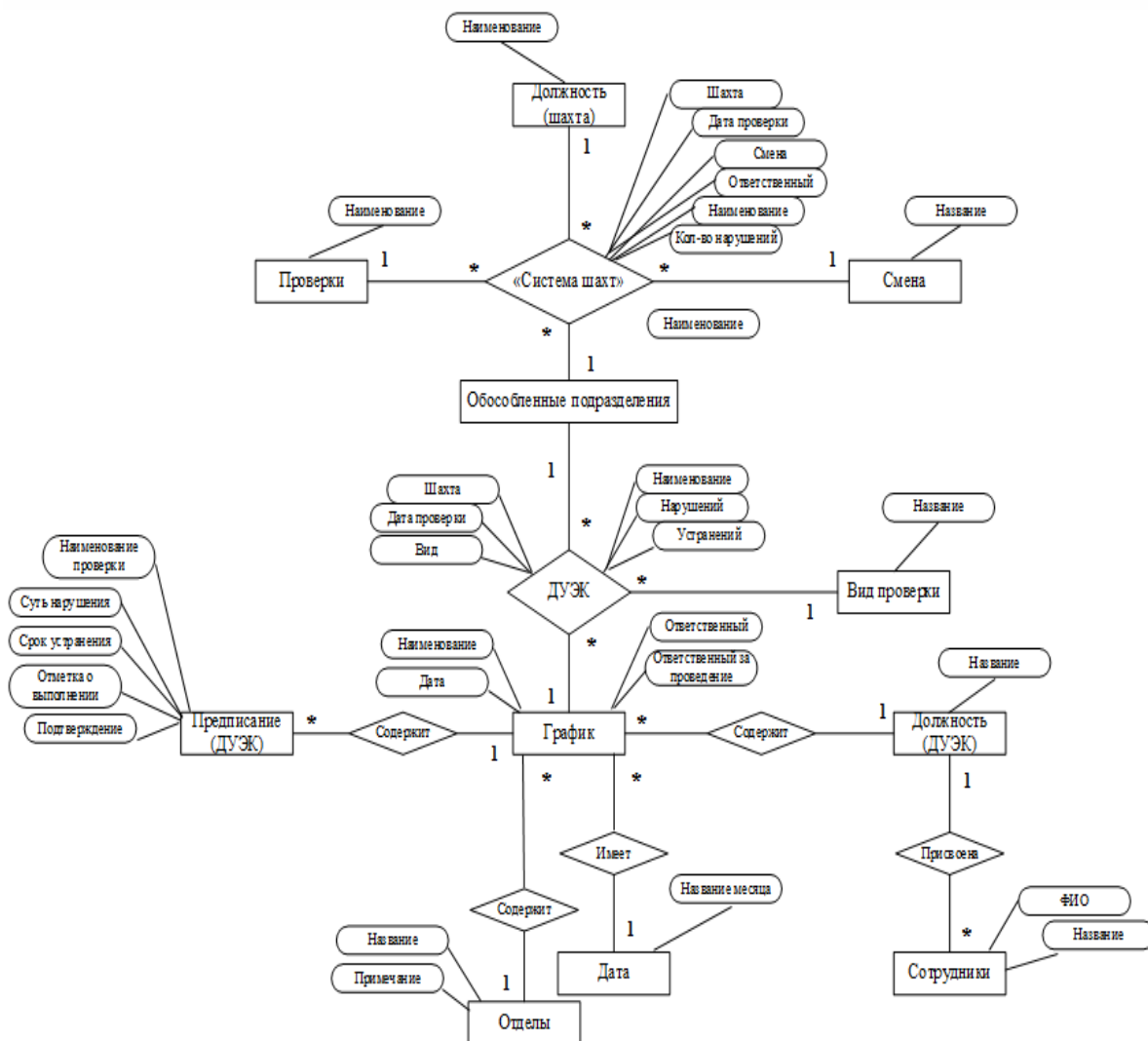


Рисунок 5 – Диаграмма «сущность-связь» системы «Контроль и мониторинг состояния охраны труда и промышленной безопасности»

Ядром любой базы является модель представления данных, т.е. их информационная структура. Основная информационная модель данных на основе записей, которая будет использоваться в разрабатываемой системе – реляционная модель.

Реляционная модель представляет собой базу данных в виде множества взаимосвязанных отношений. В каждой связи одно отношение может выступать как основное, а другое отношение выступает в роли подчиненного.

Таким образом, один кортеж основного отношения может быть связан с несколькими кортежами подчиненного отношения. Для поддержки этих связей оба отношения должны содержать наборы атрибутов, по которым они связаны. В основном отношении – это первичный ключ отношения, который

однозначно определяет кортеж основного отношения [7].

Разрабатываемая система «Контроль и мониторинг состояния охраны труда и промышленной безопасности» основывается на реляционной модели данных, которая представлена на рисунке 6.

Таким образом, разработанная база данных нормализована и соответствует четырем нормальным формам, так как каждая таблица имеет первичный ключ, каждое поле таблицы представляет уникальный тип информации, все поля атомарны (1НФ), каждое поле таблицы функционально зависит от первичного ключа, который идентифицирует исходный объект таблицы (2НФ), а таблица не содержит транзитивных зависимостей, т.е. столбцы, не являющиеся ключевыми, зависят

от первичного ключа таблицы и не зависят от всех остальных столбцов (ЗНФ).

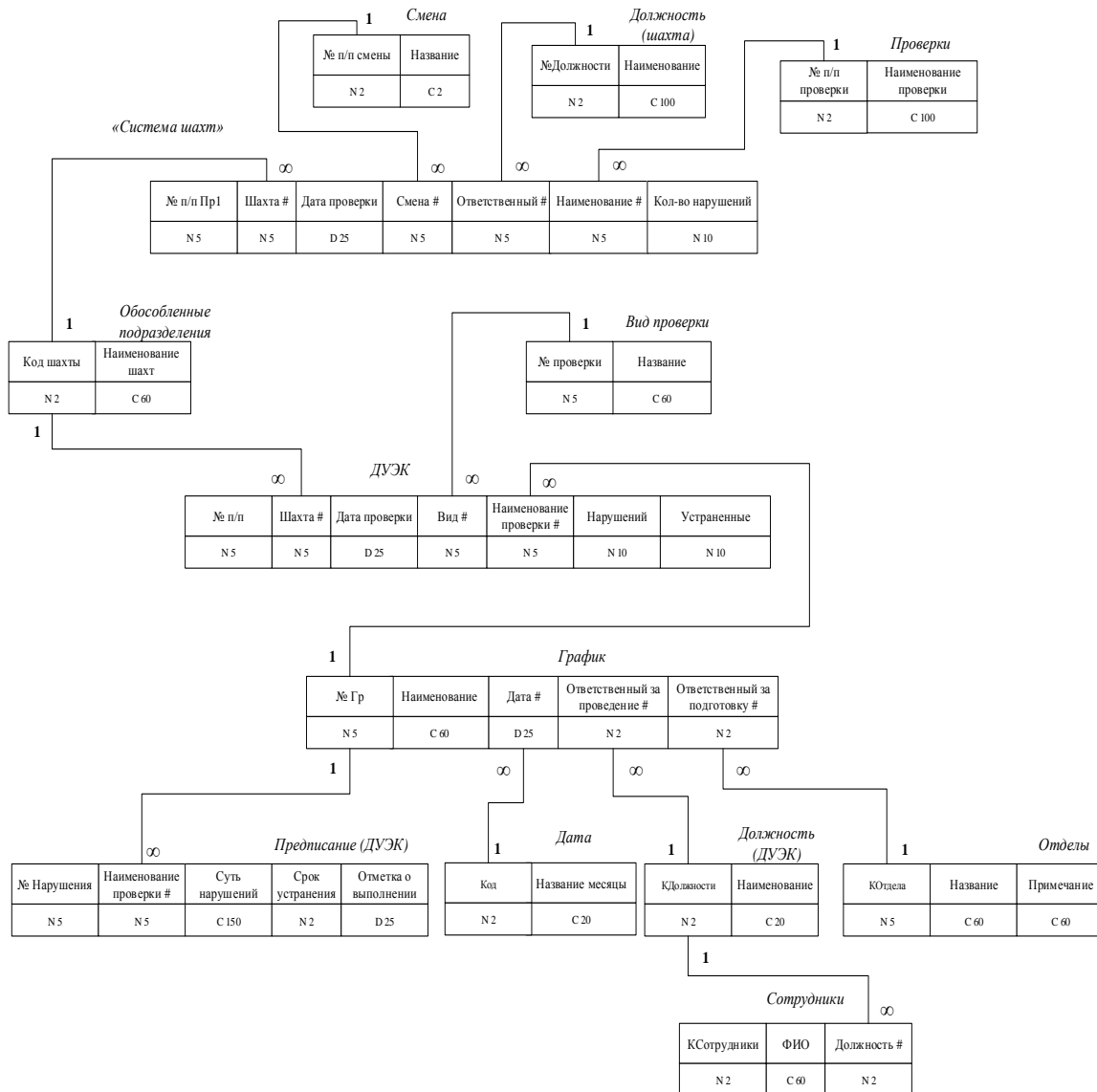


Рисунок 6 – Реляционная модель данных системы «Контроль и мониторинг состояния охраны труда и промышленной безопасности»

Программная реализация системы контроля и мониторинга состояния охраны труда и промышленной безопасности

Платформой для разрабатываемой системы был выбран программный продукт Microsoft Access, т.к. специалисты службы и отдела ОТ используют программный пакет Microsoft Office.

В проектируемой системе для работы созданы 14 таблиц, 22 экранные формы, 17 отчетов, более 20 запросов и макросов.

Структура реляционной базы данных в Access задается схемой данных (см. рис. 7), которая имеет иерархическую структуру и называется канонической реляционной моделью предметной области. Связи между таблицами

устанавливаются в соответствии с проектом логической структуры базы данных. Схема данных в Access является не только средством графического отображения базы, она активно используется системой в процессе обработки данных.

Система обеспечена защитой двумя самыми распространенными способами: установка пароля и разделение информации на уровне пользователей. В разрабатываемой системе «КМСОТ и ПБ» три основные группы пользователей – шахты, компания и Министерство. Никто кроме специалистов отдела «Донецкая угольная энергетическая компания» не имеет право осуществить резервное копирование базы, такое разделение информации необходимо для защиты от

несанкционированной утечки информации, последствия, которая может повлечь за собой необратимые

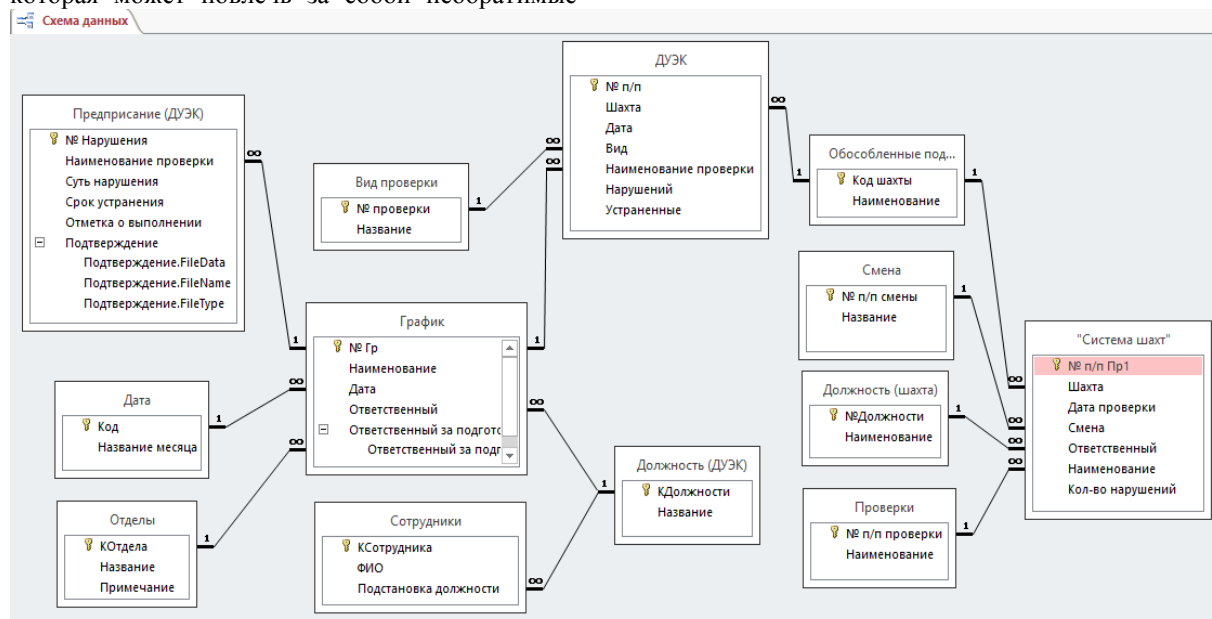


Рисунок 7 – Схема данных системы «КМСОТ и ПБ»

Выводы

Анализ угледобывающих предприятий показал, что обеспечение безопасности и безвредности производства, профилактика травматизма и заболеваний, связанных с неблагоприятными условиями труда одна из ведущих проблем на предприятиях такого рода. Разработанная система контроля и мониторинга состояния охраны труда и промышленной безопасности позволяет минимизировать время на принятие управленческих решений, что приводит к снижению уровня производственного травматизма.

Литература

1. Ефремова О. С. Охрана труда в организации в схемах и таблицах / О. С. Ефремова. – М. : Альфа-Пресс, 2012. – 108 с.
2. Ефремова О. С. Охрана труда от А до Я: Практическое пособие / О. С. Ефремова. – М. : Альфа-Пресс, 2013. – 672 с.
3. Резникова Г. А. Разработка системы контроля и мониторинга состояния промышленной безопасности на угледобывающих предприятиях / Г. А. Резникова, И. А. Тарасова // Информатика, управляющие системы, математическое и компьютерное моделирование в рамках III форума «Инновационные перспективы Донбасса» (ИУСМКМ – 2017). – Донецк : ДонНТУ, 2017. – С. 764-769.
4. Асланов М. Системный анализ и принятие решений в деятельности учреждений реального сектора экономики, связи и транспорта / М. Асланов, А. Шатраков. – М. : Экономика, 2010. – 406 с.
5. Беляков Г. И. Безопасность жизнедеятельности. Охрана труда: Учебник для бакалавров / Г. И. Беляков. – М. : Юрайт, 2013. – 572 с.
6. Тимченко Т. Н. Системный анализ в управлении: Учебное пособие / Т. Н. Тимченко. – М. : ИД РИОР, 2013. – 161 с.
7. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт, Ю. Г. Гордиенко, А. В. Слепцов – М. : Вильямс, 2001. – 1072 с.
8. Анфилатов В. С. Системный анализ в управлении / В. С. Анфилатов, А. А. Емельянов, А. А. Кукушкин. – М. : ФиС, 2009. – 368 с.
9. Громова Е. Н. Системный анализ информационных комплексов: Учебное пособие / Е. Н. Громова. – СПб. : Лань, 2016. – 336 с.
10. Данелян Т. Я. Теория систем и системный анализ: Учебно-методический комплекс / Т. Я. Данелян. – М. : Ленанд, 2016. – 360 с.
11. Пирогов В. Ю. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В. Ю. Пирогов. – СПб. : БХВ-Петербург, 2009. – 528 с.

Ирина Александровна Тарасова, Галина Анатольевна Резникова *Разработка системы контроля и мониторинга состояния охраны труда и промышленной безопасности на угледобывающих предприятиях.* Выполнен анализ угледобывающих предприятий, на основании которого разработана система контроля и мониторинга состояния охраны труда и промышленной безопасности на шахтах государственного предприятия «Донецкая угольная энергетическая компания», позволяющая минимизировать время на принятие управленческих решений, что приводит к снижению уровня производственного травматизма.

Ключевые слова: система контроля и мониторинга, промышленная безопасность, травматизм, база данных, угледобывающие предприятия

Tarasova Irina, Reznikova Galina. *Development of the system of control and monitoring the state of labor protection and industrial safety in coal-mining enterprises.* **The Abstract.** The analysis of coal-mining enterprises on the basis of which the system of control and monitoring of the state of labor protection and industrial safety at the mines of the state enterprise "Donetsk Coal Energy Company" was developed, which allows to minimize the time for making managerial decisions, which leads to a reduction in the level of industrial injuries.

Key words: control and monitoring system, industrial safety, injuries, database, coal mining enterprises

Статья поступила в редакцию 20.09.2017
Рекомендована к публикации д-ром техн. наук В.Н. Павлышом

CONTENT

| | |
|---|-----|
| Andryukhin A.I. Probabilistic automata and asynchrony..... | 5 |
| Anopriyenko A. Generalizations of Moore's Law. | 14 |
| Babich K., Sekirin A., Novikov D. Subsystem for optimizing the operation of flexible production systems using multicriteria genetic algorithms | 24 |
| Belkov D.V. Methods and computer structures for optimization of file allocation in the computer networks | 29 |
| Bodnar AV, Nikolaenko DV, Fuzzy Logic Model in the Functional Algorithm of Information Management and Enterprise Communications..... | 37 |
| Gulyaev V.A., Orlov J. C. Application of decision criteria and game theory in economic index management of a metallurgical facility. | 43 |
| A.A. Dvoykin, M.V. Privalov, A.I.Voronova Architecture and distributed medical image processing scenarios of the Web-oriented PACS..... | 50 |
| Dyachenko O. N., Zinchenko Y.E., Dyachenko V. O. Application of methods of antinoise coding for compact testing of digital circuits. | 55 |
| Zinchenko Y. E., Lench V.S., Zinchenko T.A., Pavlysh V.N.. Simulation of peripheral devices in the system of remote access to FPGA-complexes. | 60 |
| Ivanitsa S. Designing of computer components for realization of tetralogic operations. | 69 |
| Iskra O.A., Model of Decision-Making Support for a Mobile Application Life Cycle Management | 77 |
| Malcheva R.V., Zavadsкая T.V., Kudoyar V.I. An investigation of a graphical system with a vertical architecture for searching for intersection of a ray with a group of polygonal objects | 83 |
| Martynenko T.V., Shcherbakov A.A., Bodnar B.V. Analysis recognizing methods of the emotional state of a person to detect abnormal behavior in video analytics systems. | 89 |
| Matiakh I.V., Savkova E.O., Chengar O.V. Investigation of the problem of choosing a minimum set of services for expanding the activity of an enterprise using the example of an Internet provider | 94 |
| Minenko A. S., Radevich E. V. Approximate analysis of convective Stefan problem. | 100 |
| Olshevsky A.I. Training system for working with container classes..... | 106 |
| Svetlichnaya V.A., Andrievskaya N.K., Chalenko K.Y. Development of the functional structure of the logistics system of formation of orders for the online store.. | 111 |
| Tarasova Irina, Reznikova Galina. Development of the system of control and monitoring the state of labor protection and industrial safety in coal-mining enterprises.. | 119 |
| Content..... | 128 |

Научное издание

ИНФОРМАТИКА И КИБЕРНЕТИКА

Научный журнал

(на русском, английском языках)

№ 3(9)-2017

Ответственный за выпуск А.И. Андриухин

Технические редакторы Т.В. Завадская, А.В. Звягинцева

Компьютерная верстка Д.А. Филипишин

Подписано к выпуску 27.10.2017. Усл. печ. лист. 10. Уч.- изд. лист. 7,1.

Адрес редакции: ДНР, 83001, г. Донецк, ул. Артема, 58, ГОУ ВПО «ДонНТУ», 5-й учебный корпус, к. 425. Тел.: +38 (062) 301-08-51 E-mail: infcyb.donntu@yandex.ru, URL: <http://infcyb.donntu.org>