

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25

2019

№ 8

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

САПР

КОМПЬЮТЕРНАЯ ГРАФИКА

МЕТОДЫ ПРОГРАММИРОВАНИЯ

ОПЕРАЦИОННЫЕ СИСТЕМЫ И СРЕДЫ

ТЕЛЕКОММУНИКАЦИИ
И ВЫЧИСЛИТЕЛЬНЫЕ СЕТИ

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

НЕЙРОСЕТИ И
НЕЙРОКОМПЬЮТЕРЫ

СТРУКТУРНЫЙ СИНТЕЗ

ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ
СИСТЕМЫ

ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

ОПТИМИЗАЦИЯ И МОДЕЛИРОВАНИЕ

ИТ В ОБРАЗОВАНИИ

ГИС

Рисунки к статье Ш. С. Фахми, Н. Г. Рыжова, Х. А. Хасана, Е. С. Калининой

«АДАПТИВНЫЙ АЛГОРИТМ КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ИЗОБРАЖЕНИЙ В ТРАНСПОРТНЫХ СИСТЕМАХ НАБЛЮДЕНИЯ»

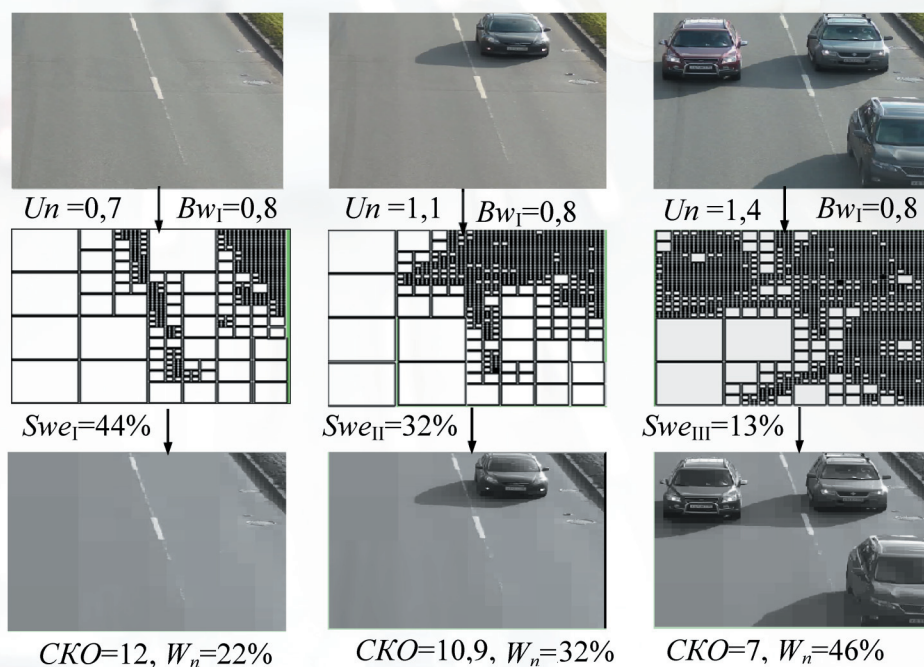


Рис. 2. Пример изменения сетки ДКП для изображений класса $Bw_I - Un_1$, (Un – мера неопределенности; Bw – мера широкомасштабности)

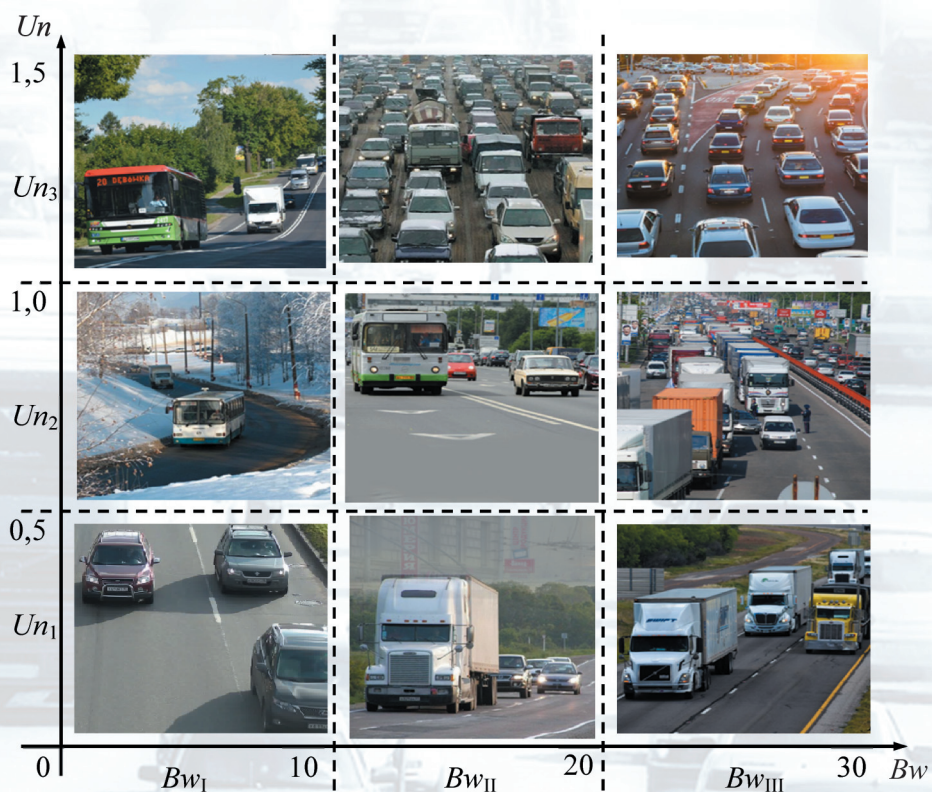


Рис. 3. Классификация и выборка транспортных сюжетов

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Том 25
2019
№ 8

ТЕОРЕТИЧЕСКИЙ И ПРИКЛАДНОЙ НАУЧНО-ТЕХНИЧЕСКИЙ ЖУРНАЛ

Издается с ноября 1995 г.

DOI 10.17587/issn.1684-6400

УЧРЕДИТЕЛЬ

Издательство "Новые технологии"

СОДЕРЖАНИЕ

МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ

- Родионов А. Н.** Абстрактные роли и примитивы ролевого моделирования сущностей в системе "концептуальная—логическая—физическая модели данных" 451
- Зак Ю. А.** Алгоритмы оперативного перераспределения ресурсов в системах сетевого планирования и управления выполнением проектов 467

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

- Фахми Ш. С., Рыжов Н. Г., Хасан Х. А., Калинина Е. С.** Адаптивный алгоритм кодирования и декодирования изображений в транспортных системах наблюдения 475

ПРИКЛАДНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ

- Гвоздев В. Е., Христовуло О. И.** Информационная поддержка принятия решений в задачах управления отходами на основе математико-геоинформационного моделирования 482

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

- Вишнеков А. В., Иванова Е. М., Басова К. Э., Ветелина Е. О.** Интерактивный учебно-исследовательский комплекс для моделирования процессов в вычислительных системах 490

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И ПРОИЗВОДСТВЕ

- Грибова В. В., Величко А. С., Колмогоров А. В.** Концепция цифровой платформы для производственной деятельности 502

Главный редактор:

СТЕМПКОВСКИЙ А. Л.,
акад. РАН, д. т. н., проф.

Зам. главного редактора:

ИВАННИКОВ А. Д., д. т. н., проф.
ФИЛИМОНОВ Н. Б., д. т. н., с.н.с.

Редакционный совет:

БЫЧКОВ И. В., акад. РАН, д. т. н.
ЖУРАВЛЕВ Ю. И.,
акад. РАН, д. ф.-м. н., проф.
КУЛЕШОВ А. П.,
акад. РАН, д. т. н., проф.
ПОПКОВ Ю. С.,
акад. РАН, д. т. н., проф.
РУСАКОВ С. Г.,
чл.-корр. РАН, д. т. н., проф.
РЯБОВ Г. Г.,
чл.-корр. РАН, д. т. н., проф.
СОЙФЕР В. А.,
акад. РАН, д. т. н., проф.
СОКОЛОВ И. А.,
акад. РАН, д. т. н., проф.
СУЕТИН Н. В., д. ф.-м. н., проф.
ЧАПЛЫГИН Ю. А.,
акад. РАН, д. т. н., проф.
ШАХНОВ В. А.,
чл.-корр. РАН, д. т. н., проф.
ШОКИН Ю. И.,
акад. РАН, д. т. н., проф.
ЮСУПОВ Р. М.,
чл.-корр. РАН, д. т. н., проф.

Редакционная коллегия:

АВДОШИН С. М., к. т. н., доц.
АНТОНОВ Б. И.
БАРСКИЙ А. Б., д. т. н., проф.
ВАСЕНИН В. А., д. ф.-м. н., проф.
ВАСИЛЬЕВ В. И., д. т. н., проф.
ВИШНЕКОВ А. В., д. т. н., проф.
ДИМИТРИЕНКО Ю. И., д. ф.-м. н., проф.
ДОМРАЧЕВ В. Г., д. т. н., проф.
ЗАБОРОВСКИЙ В. С., д. т. н., проф.
ЗАРУБИН В. С., д. т. н., проф.
КАРПЕНКО А. П., д. ф.-м. н., проф.
КОЛИН К. К., д. т. н., проф.
КУЛАГИН В. П., д. т. н., проф.
КУРЕЙЧИК В. В., д. т. н., проф.
ЛЬВОВИЧ Я. Е., д. т. н., проф.
МАРТЫНОВ В. В., д. т. н., проф.
МИХАЙЛОВ Б. М., д. т. н., проф.
НЕЧАЕВ В. В., к. т. н., проф.
ПОЛЕЩУК О. М., д. т. н., проф.
САКСОНОВ Е. А., д. т. н., проф.
СОКОЛОВ Б. В., д. т. н., проф.
ТИМОНИНА Е. Е., д. т. н., проф.
УСКОВ В. Л., к. т. н. (США)
ФОМИЧЕВ В. А., д. т. н., проф.
ШИЛОВ В. В., к. т. н., доц.

Редакция:

БЕЗМЕНОВА М. Ю.

Информация о журнале доступна по сети Internet по адресу <http://novtex.ru/IT>.
Журнал включен в систему Российского индекса научного цитирования и базу данных RSCI на платформе Web of Science.

Журнал входит в Перечень научных журналов, в которых по рекомендации ВАК РФ должны быть опубликованы научные результаты диссертаций на соискание ученой степени доктора и кандидата наук.

INFORMATION TECHNOLOGIES

INFORMACIONNYYE TEHNOLOGII

Vol. 25
2019
No. 8

THEORETICAL AND APPLIED SCIENTIFIC AND TECHNICAL JOURNAL

Published since November 1995

ISSN 1684-6400

CONTENTS

MODELING AND OPTIMIZATION

Rodionov A. N. The Abstract Roles and the Primitives of Role Modeling in The Conceptual, Logical and Physical Data Models System 451

Zack Yu. A. Algorithms for the Operational Redistribution of Resources in Network Planning and Project Management Systems 467

DIGITAL PROCESSING OF SIGNALS AND IMAGES

Fahmi S. Sh., Ryjov N. G., Hasan Y. A., Kalinina E. S. Adaptive Algorithm of Image Encoding and Decoding in Transport Video Surveillance Systems 475

APPLICATION INFORMATION SYSTEMS

Gvozdev V. E., Khristodulo O. I. Information Support of Decision-Making in Waste Management Problems on the Basis of Mathematical and Geoinformation Modeling 482

INFORMATION TECHNOLOGIES IN EDUCATION

Vishnekov A. V., Ivanova E. M., Basova K. E., Vetelina E. O. Interactive Educational-Research Complex for Processes Simulation in Computer Systems 490

INFORMATION TECHNOLOGY IN THE ECONOMY AND PRODUCTION

Gribova V. V., Velichko A. S., Kolmogorov A. V. Digital Platform Concept for Business Activities 502

Editor-in-Chief:

Stempkovsky A. L., Member of RAS,
Dr. Sci. (Tech.), Prof.

Deputy Editor-in-Chief:

Ivannikov A. D., Dr. Sci. (Tech.), Prof.
Filimonov N. B., Dr. Sci. (Tech.), Prof.

Chairman:

Bychkov I. V., Member of RAS,
Dr. Sci. (Tech.), Prof.
Zhuravljov Yu. I., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Kuleshov A. P., Member of RAS,
Dr. Sci. (Tech.), Prof.
Popkov Yu. S., Member of RAS,
Dr. Sci. (Tech.), Prof.
Rusakov S. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Ryabov G. G., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Soifer V. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Sokolov I. A., Member of RAS,
Dr. Sci. (Phys.-Math.), Prof.
Suetin N. V.,
Dr. Sci. (Phys.-Math.), Prof.
Chaplygin Yu. A., Member of RAS,
Dr. Sci. (Tech.), Prof.
Shakhnov V. A., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.
Shokin Yu. I., Member of RAS,
Dr. Sci. (Tech.), Prof.
Yusupov R. M., Corresp. Member of RAS,
Dr. Sci. (Tech.), Prof.

Editorial Board Members:

Avdoshin S. M., Cand. Sci. (Tech.), Ass. Prof.
Antonov B. I.
Barsky A. B., Dr. Sci. (Tech.), Prof.
Vasenin V. A., Dr. Sci. (Phys.-Math.), Prof.
Vasiliev V. I., Dr. Sci. (Tech.), Prof.
Vishnekov A. V., Dr. Sci. (Tech.), Prof.
Dimitrienko Yu. I., Dr. Sci. (Phys.-Math.), Prof.
Domrachev V. G., Dr. Sci. (Tech.), Prof.
Zaborovsky V. S., Dr. Sci. (Tech.), Prof.
Zarubin V. S., Dr. Sci. (Tech.), Prof.
Karpenko A. P., Dr. Sci. (Phys.-Math.), Prof.
Kolin K. K., Dr. Sci. (Tech.)
Kulagin V. P., Dr. Sci. (Tech.), Prof.
Kureichik V. V., Dr. Sci. (Tech.), Prof.
Ljvovich Ya. E., Dr. Sci. (Tech.), Prof.
Martynov V. V., Dr. Sci. (Tech.), Prof.
Mikhailov B. M., Dr. Sci. (Tech.), Prof.
Nechaev V. V., Cand. Sci. (Tech.), Ass. Prof.
Poleschuk O. M., Dr. Sci. (Tech.), Prof.
Saksonov E. A., Dr. Sci. (Tech.), Prof.
Sokolov B. V., Dr. Sci. (Tech.)
Timonina E. E., Dr. Sci. (Tech.), Prof.
Uskov V. L. (USA), Dr. Sci. (Tech.)
Fomichev V. A., Dr. Sci. (Tech.), Prof.
Shilov V. V., Cand. Sci. (Tech.), Ass. Prof.

Editors:

Bezmenova M. Yu.

Complete Internet version of the journal at site: <http://novtex.ru/IT>.

According to the decision of the Higher Certifying Commission of the Ministry of Education of Russian Federation, the journal is inscribed in "The List of the Leading Scientific Journals and Editions wherein Main Scientific Results of Theses for Doctor's or Candidate's Degrees Should Be Published"

А. Н. Родионов, д-р техн. наук, вед. науч. сотр., e-mail: ran@newmail.ru,
Вычислительный центр ДВО РАН, г. Хабаровск

Абстрактные роли и примитивы ролевого моделирования сущностей в системе "концептуальная—логическая—физическая модели данных"

Ролевая концепция достаточно часто используется при построении концептуальных моделей данных. С ее помощью, в частности, появляется возможность динамически выделять подмножества сущностей, которым разрешено участвовать в тех или иных взаимодействиях. Воплощение ролевого поведения на следующем — логическом — уровне моделирования может достигаться либо многократным клонированием сущностных типов по одному клону на каждую роль, либо идти по пути создания специализированных ролевых типов и закрепления последних за определенными типами сущностей. При этом оба подхода порождают сопутствующие проблемы, которые, как в первом случае, приводят к неконтролируемому росту клоновых объектов логических моделей, или, как во втором, требуют использования ролевых констант для обращения к сущностям, играющих определенные роли.

Концепция абстрактных ролей, предложенная и реализованная в настоящей работе, устраняет указанные недостатки ролевых типов, ограничиваясь использованием исключительно только той информации, которую содержат логические модели. Разработан универсальный шаблон ролевого моделирования с единственным абстрактным ролевым типом, содержащим список абстрактных ролей. Список постоянен для выбранной конфигурации логической модели. Шаблон включает реальные ролевые типы, каждый из которых привязан к собственному сущностному типу. Такая организация также решает проблему формирования виртуального множества сущностей, играющих одну и ту же роль, но принадлежащих разным сущностным типам.

Ключевые слова: абстрактные роли, ролевое моделирование, ролевые паттерны

Введение

За последние три десятилетия усилиями многочисленных исследователей все скольконибудь значимые аспекты ролевого моделирования, включая назначение, функции и представление ролей в концептуальных и логических моделях (КМ и ЛМ), были досконально изучены и нашли отражение во множестве публикаций. Проанализировав содержание более чем 20 работ, Ф. Стейманн [1] идентифицировал 15 ролевых характеристик (свойств) и выделил три потенциальных способа представления ролей в моделях данных: посредством именованых отношений, возникающих между объектами [2—4]; придания ролевой семантической окраски объектам, полученным в результате проведения процедур специализации и/или генерализации над множествами, образованными экземплярами исходных объектов [5—7]; простым перечислением экземпляров ролей в специализированных ролевых

объектах и установления соответствия между последними и не ролевыми объектами [8—10].

Ролевая компонента, представленная определенным подмножеством ролевых свойств и использующая один из трех способов представления, присутствует сегодня во всех широко известных и распространенных методологиях моделирования: на основе модели ER (модель "сущность—связь") [4] и ее расширений [7, 11—13]; ORM — объектно-ролевым моделированием [2, 14, 15]; UML — унифицированном языке моделирования [3, 16, 17]; OntoUML — онтологическом UML [18], унифицированной метамодели, интегрирующей ER, ORM и UML [17]. Наблюдаемые различия свидетельствуют об отсутствии консенсуса в вопросах, касающихся прежде всего определения понятия роли, а также возложенных на нее функций. Но даже не этот аспект следует считать определяющим. Восприятие роли зависит от рамочных основ, формирующих каркас каждой из методологий. OntoUML, например, строится

на базе развитой системы объектных типов (категорий), включающей непосредственно две ролевые категории: *Role* и *RoleMixin*. В отличие от OntoUML ORM организована в терминах объектов, играющих роли, где роль считается обязательным атрибутом любого отношения (взаимодействия, ассоциации).

В настоящей работе также, как и в OntoUML, все типы (именованные множества) классифицированы. В первую очередь проводится различие между сущностными и несущностными типами. В первых сосредоточены сведения о сущностях, из которых, собственно говоря, и состоит любая предметная область (ПрО). Вторые являются производными от первых и содержат объекты, выполняющие вспомогательную, обеспечивающую по отношению к сущностям функцию. По мере изложения различные категории несущественных типов будут последовательно появляться, объясняться и описываться в работе.

Постулируется, что роль полностью задает подмножество сущностей, участвующих в том или ином взаимодействии, и является одной из обязательных характеристик взаимодействия. Заметим, что подобная трактовка назначения роли подразумевает закрепление за ней еще одной важной для моделирования данных функции — функции вторичной, динамической подтипизации, обеспечивающей отбор из сущностных типов только тех экземпляров, которые в состоянии играть ту или иную роль. (Аналогичную функцию выполняют и два других известных вида подтипизации — фазная и категорийная подтипизации, которые в работе не рассматриваются.) Укажем, что ролевую подтипизацию от фазной и категорийной отличает то, что "ролевые" множества могут пересекаться.

Базовый, первичный вид подтипизации, обозначенный нами как характеристический, выполняется в начальной фазе концептуального моделирования и формирует каркас модели в виде системы сущностных типов, экземпляры которых не пересекаются. Собственно говоря, наличие такого каркаса, на который вынуждена опираться ролевая подтипизация, является источником одной из ключевых проблем ролевого моделирования, которая проявляется уже на логическом уровне. Обозначенная проблема, в частности, сводится к поиску способа представления неограниченной совокупности ролевых подмножеств в логических моделях, элементами которых к тому же могут быть экземпляры, принадлежащие разным типам.

Другая актуальная задача, которая на первый взгляд может показаться тривиальной, состоит в том, что в случае использования специализированных ролевых таблиц, которые содержат исключительно названия ролей, а не сущности, играющие определенные роли, мощность и содержание таких таблиц становятся известными только в процессе формирования базы данных. Но чтобы обеспечить соблюдение принципа независимости программ и данных, экземпляры ролей должны быть известны уже на этапе проектирования. В противном случае просто невозможно выделить подмножества сущностей, соответствующие той или иной роли. Устранение противоречия между непредсказуемостью и известностью реальных ролей на этапе проектирования, а также решение проблемы смешивания экземпляров, принадлежащих разным типам, но играющих одну и ту же роль, составляют основное содержание настоящей работы. В частности, в статье излагаются решения, основанные на использовании абстрактных ролей, число и состав которых постоянен для любой предметной области.

В название работы вынесено словосочетание "физическая модель", которое призвано подчеркнуть необходимость комплексного решения задач моделирования, не ограниченного рамками только формальных концептуальной и логической моделей. "Физическая" означает, что данных, размещенных в таблицах баз данных и структурированных в соответствии с логической моделью, должно быть достаточно для динамического формирования экземпляров ролевых подмножеств в процессе обращения к этим данным из прикладных программных сред.

1. Категории объектных типов предметных областей и концептуальных моделей данных

Ролевые поведения демонстрируют далеко не все категории типов, которые принято различать в ПрО. Если основываться на том, что ПрО состоит из сущностей, то роли будут характерны только для сущностей. Поэтому вопрос о том, какие типы относить к сущностям, а какие нет — ключевой для ролевого моделирования. Будем опираться на подходы, явно классифицирующие типы в соответствии с их назначением и поведением, которые они демонстрируют в моделируемых универсумах. Ввиду обширности, неоднозначности и многогранности исследований, посвященных различным

аспектам типизации объектов, ограничимся кратким изложением состояния этого вопроса.

Категорийное, шаблонное моделирование сегодня — один из трендов в концептуальном моделировании данных, позволяющий ускорять и упрощать процесс проектирования. Разные исследователи выделяют разные категории объектных типов. Сошлемся на ряд заметных результатов, полученных в этой области. Обширную и развитую систему типов, которая включает такие виды, как *Kinds*, *Subkinds*, *Roles*, *RolesMixins*, *Phases*, *Categories* и *Mixins*, обосновал и предложил в рамках унифицированной фундаментальной онтологии (UFO) Ж. Гуиззарди [18], она получила дальнейшее развитие в лице MLT — многоуровневой теории [19]. Все перечисленные типы для их индустриальной поддержки в последующем в виде профайла были встроены в UML, а также использованы для разработки многочисленных шаблонов концептуального моделирования.

Различные категории типов обнаруживаются и в другой популярной методологии — ORM [2], несмотря на то что на их обособлении не делается явного акцента. В отдельные категории сведены нелексические (сущности — *Entities*) и лексические (значения — *Values*) типы, некоторые ассоциации, трактуемые как самостоятельные типы (*Objectified Associations*), типы категоризации (*Categorization Types*), коллекционные типы (*Collection Types*). Аналоги "*Objectified Associations*" присутствуют и в работе [18], но там они получили более компактное название — "*Relators*", которого и будем придерживаться далее.

В работе [20] показано, что для комплексного моделирования и представления сущностей, принадлежащих одному и тому же типу, требуется в общем случае сформировать сущностный кластер, представленный такими взаимосвязанными сущностными подтипами, как *Patterns*, *Samples* и *Instances*.

По сути, каждая категория (класс) типов демонстрирует собственное, отличное от других категорий поведение. Но все авторы упомянутых работ выделяют сущностные типы в качестве базовых типов. Дать исчерпывающее определение сущности и сформировать критерий, на основании которого можно было бы определить — относится ли объект к сущности или нет, оказалось весьма непросто. Данный вопрос до сих пор остается открытым, и многие исследователи стараются обойти его стороной, в том числе ссылаясь либо на очевидность или тождественность объектов и

сущностей, либо используя общие формулировки наподобие следующих: сущность есть "некоторый объект, представляющий интерес для организации" [21]; сущность — это "нечто, принадлежащее объективной реальности, обличенное в материальную форму" [22]; "сущность — это вещь, которая может быть однозначно идентифицирована" [4].

"Сущностный" вопрос — это вопрос, касающийся в большей степени не информатики, а скорее лингвистики и философии. Не претендуя на универсальность и какую-либо общность, выделим два приема практической направленности, которые могут оказать помощь в идентификации сущностных объектов. Например, если мысленно зафиксировать время, то в ПрО останутся только сущности. Останутся процессы, часто воспринимаемые как сущности, исчезнут ролевые и фазовые объекты, не станет семантических объектов и объектов категоризации. Другой способ выделить сущности в массе объектов — последовательно исключать все другие категории до тех пор, пока в остатке не останутся сущности.

Все дальнейшее изложение будет касаться описания поведения и представления на концептуальном, логическом и физическом уровнях моделирования пар "тип сущности—роли".

2. Роль как объект моделирования

Роль, если рассматривать ее как лингвистическую категорию, можно соотнести с некоторым набором объектов и двумя подмножествами действий, которые эти объекты в состоянии совершать, воздействуя на другие объекты или на самих себя, либо, напротив, подвергаться действиям со стороны других объектов, в том числе также оказывая воздействия на самих себя. "Пилот", "врач", "преподаватель", "военнослужащий" — примеры ролей, соотнесенных с объектами такого типа, как "личности".

Представленная ролевая трактовка влечет за собой необходимость в дополнительной конкретизации отношений, возникающих как между сущностями в ПрО, так и между сущностями и другими видами объектов. (Будем в дальнейшем ссылаться на такие взаимодействия как на ролевые и, соответственно, использовать прилагательное "ролевой" для конкретизации отдельных элементов: экземпляров типов, самих типов и свойств экземпляров, имеющих отношение к взаимодействиям подобного рода.) Помимо действий, выполняе-

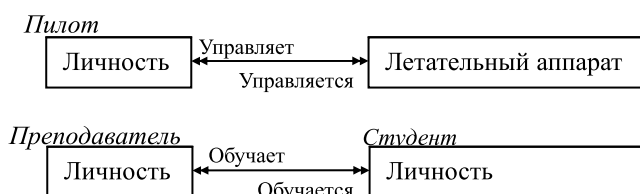


Рис. 1. Пример ассоциаций концептуальной модели, содержащих ссылки на роли

мых сущностями в ассоциации, потребуется указывать и их роли. Для этого ссылку на роль достаточно "прикрепить" к соответствующему типу сущности (рис. 1), экземплярам которой разрешено вступать в то или иное взаимодействие с экземплярами других типов ПрО.

Справедливости ради отметим, что это не единственный способ представления ролей на концептуальном уровне. Д. Кабот и Р. Равентос в работе [23] приводят и описывают еще четыре потенциальных варианта. Но все эти варианты в той или иной степени предполагают задействование произвольного числа ролевых типов, что идет вразрез с поставленной ранее задачей ограничения их числа.

Очевидно, что конкретные роли в состоянии играть только определенные подмножества тех же "личностей", а элементы, входящие в состав этих подмножеств, должны обладать какими-то уникальными характеристиками. Например, чтобы быть "пилотом" или "преподавателем", нужно соответствовать отличительным квалификационным требованиям.

Из сказанного следует, что роль одновременно задает и круг объектов, обладающих определенными характеристиками, и список действий, выполняемых этими объектами

(или перечень действий, направленных на эти объекты), и спектр отношений, в которых "ролевые" объекты могут принимать участие.

Таким образом, чтобы формально задать (описать) некоторую роль, достаточно:

- указать отношения, к которым она будет прикреплена;
- перечислить типы сущностей и в каждом типе выделить подмножество экземпляров, которые в состоянии играть эту роль;
- каждой паре <тип сущности, тип отношения> поставить в соответствие или множество действий, которые выполняют сущности в этом отношении, или множество действий, которым они подвергаются. (Последний аспект ввиду его очевидности и однозначности вынесен за рамки обсуждения.)

Отношения между перечисленными категориями, имеющими непосредственное отношение к роли, представлены на рис. 2. Здесь: R — множество ролей; E — множество сущностей; C — множество отношений, один из элементов которых обязательно тип сущности; $E_{r_z}^n$ — n -е подмножество сущностей, которые потенциально могут играть r_z -ю роль (Одну и ту же роль могут выполнять объекты разной природы, сгруппированные по этой причине в отдельные, непересекающиеся друг с другом подмножества $E_{r_z}^n$); $\bar{E}_{r_z}^n$ — собственное подмножество $E_{r_z}^n$, элементы которого в состоянии играть роль, $\bar{E}_{r_z}^n \subseteq E_{r_z}^n$. (Об отличиях $E_{r_z}^n$ и $\bar{E}_{r_z}^n$ — чуть ниже.)

Как было отмечено ранее, важнейшая функция роли, вытекающая из ее определения, — выделение в ПрО именованных подмножеств сущностей, которые способны играть эту роль. (Способны, однако, не означает, что сущность действительно будет участвовать в ролевых взаимодействиях.

Необходимо еще, чтобы роль была каким-то образом прикреплена к сущности.) Априорное именование ролевых подмножеств, являющееся обычной практикой в ПрО, дает основание предположить, что такие подмножества не пересекаются. Приведем два примера, взятых из практики. Роль — "объекты транспортного налога" — различает автомобили, мотосани, моторные лодки, гидроциклы и другие разновидности сущностных объектов. Другая роль "Экзаменаторы" может выполняться только или доцентами или профес-

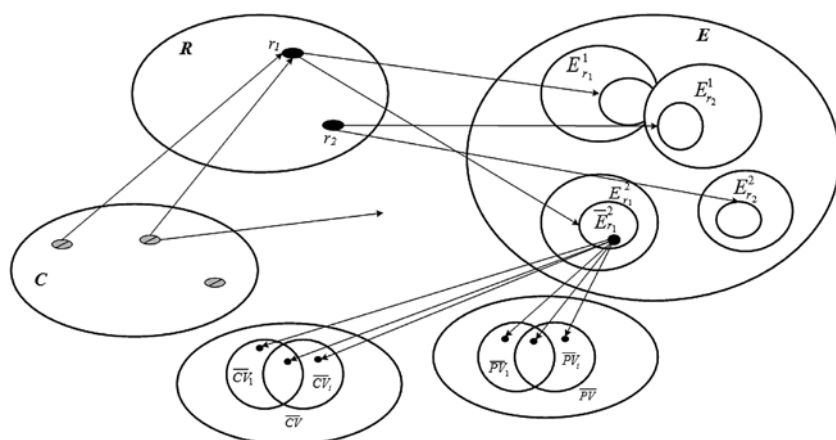


Рис. 2. Соответствия между элементами актуальных множеств, участвующих в ролевых взаимодействиях

сорами. Таким образом, для $\forall r_z$ справедливо:

$$\bigcap_{n=1 \div N} E_{r_z}^n = \emptyset.$$

В то же время стандартной является ситуация, когда сущности, принадлежащие одному типу, играют множество ролей, и поэтому их отдельные подмножества могут пересекаться. Те же моторные лодки — одновременно и объекты налогообложения, и средства передвижения.

Для многих ролей одной только ссылки на $E_{r_z}^n$ оказывается недостаточно, чтобы очертить круг "ролевых" сущностей (сущностей, задействованных во взаимодействиях, к которым прикреплены роли), потому что не все элементы $E_{r_z}^n$ в состоянии играть r_z -ю роль в конкретный момент времени. Для того чтобы отдельные экземпляры $E_{r_z}^n$ могли участвовать в подобных отношениях ("ролевых" отношениях), часть подлинных свойств этих экземпляров ("ролевых" свойств) должна иметь на этот момент времени вполне определенные значения, и/или они должны участвовать в каких-то взаимодействиях, а экземпляры этих взаимодействий, в свою очередь, также должны принимать наперед заданные значения. Например, мощность моторных лодок как объекта налогообложения должна превышать 5 лошадиных сил. Для занятия должности "профессор" необходимо иметь степень доктора наук и/или ученое звание не ниже доцента. В последнем случае — это уже экземпляры отношений, возникающих между экземплярами сущности "Личности" и объектами квалификационных типов "Ученые степени" и "Ученые звания" соответственно. (Упомянутые выше подлинные свойства — это свойства, которые, согласно работе [24], присущи экземплярам типа, изолированного от других типов.)

В связи с этим о $\bar{E}_{r_z}^n$ можно говорить как о динамическом множестве, состав которого может меняться в моменты времени t , когда происходят изменения в значениях "ролевых" свойств и/или экземплярах "ролевых" отношений.

Представим то же самое формально. Требуется задать множества $\bar{E}_{r_z}^n(t)$. Запись $\bar{E}_{r_z}^n(t)$ будет означать, что состав $\bar{E}_{r_z}^n$ фиксируется на момент времени t . Будем также полагать, что содержание $E_{r_z}^n$ априорно известно. По крайней мере, в ПрО не только принято именовать все $E_{r_z}^n$, но и предоставлять описание элементов $E_{r_z}^n$. Обозначим p_m — m -е подлинное свойство сущности $e_{r_z}^n \in E_{r_z}^n$, а $v_{p_m}(t)$ — значение этого свойства в момент времени t . Пусть \overline{PV} — множество значений подлинных свойств, которые задают r_z -ю роль для элементов множества $E_{r_z}^n$.

Другими словами, \overline{PV} — это множество индикаторов, по достижении которых $e_{r_z}^n$ переходит в категорию $\bar{e}_{r_z}^n$. В общем случае, для $\forall t$ возможны ситуации, когда $PV(t) \not\subseteq \overline{PV}$, т. е. никакие элементы $\bar{e}_{r_z}^n$ не в состоянии играть роль r_z . В простейшем варианте

$$\overline{PV} = \{ \langle p_m, [\bar{v}_{p_m}^s, \bar{v}_{p_m}^q], T_{p_m} \rangle \}_1^M, \quad (1)$$

где M — число подлинных свойств сущностей, образующих $E_{r_z}^n$; $[\bar{v}_{p_m}^s, \bar{v}_{p_m}^q]$ — диапазон значений, которое должно приобрести свойство, чтобы выполнять роль; $\bar{v}_{p_m}^s$ и $\bar{v}_{p_m}^q$ — соответственно начальное и конечное значения свойства; T_{p_m} — имя домена, на котором определено свойство. (Далее T_{p_m} как несущественное для обсуждения будем опускать.) Так как не все свойства задают ролевую принадлежность сущностей, а индекс m в (1) "пробегают" по всем свойствам, исключим "неролевые" свойства, присвоив $\bar{v}_{p_m}^s$ и $\bar{v}_{p_m}^q$ значение *null* в соответствующем триplete $\langle p_m, [\bar{v}_{p_m}^s, \bar{v}_{p_m}^q], T_{p_m} \rangle$.

Приведем пример задания \overline{PV} для множества тех же моторных лодок. $\overline{PV} = \{ \langle \text{Мощность}, 5 \rangle, \langle \text{Грузоподъемность}, 700 \rangle \}$.

Как показывают наблюдения, практика в отношении \overline{PV} не ограничивается безальтернативностью. В реальности таких множеств, как \overline{PV} , может оказаться несколько. Например:

$$\overline{PV}_1 = \{ \langle \text{Мощность}, 5 \rangle, \langle \text{Грузоподъемность}, 700 \rangle \}$$

$$\overline{PV}_2 = \{ \langle \text{Мощность}, 9 \rangle, \langle \text{Грузоподъемность}, 700 \rangle, \langle \text{Материал}, \text{ПВХ} \rangle \}.$$

Поэтому в действительности \overline{PV} — это система из l -множеств ($l = \overline{1}, \overline{L}$), задающая совокупность ролевых альтернатив:

$$\overline{PV} = \{ \overline{PV}_1, \overline{PV}_2, \dots, \overline{PV}_l, \dots, \overline{PV}_L \}.$$

В выражении (1) $\bar{v}_{p_m}^s$ и $\bar{v}_{p_m}^q$ — это константы. В то же время не исключены ситуации, когда $\bar{v}_{p_m}^s$ и $\bar{v}_{p_m}^q$ задаются посредством формул. Поэтому более корректно использовать предикатную форму для задания $\bar{v}_{p_m}^s$ и $\bar{v}_{p_m}^q$.

По аналогии с \overline{PV}_l обозначим через $PV_l^{e_{r_z}^n}(t)$ l -е множество значений подлинных свойств элемента $e_{r_z}^n$, которые он приобретает в момент времени t (ввиду громоздкости верхнего индекса в $PV_l^{e_{r_z}^n}(t)$ последний будем опу-

скать, полагая, что $PV_f(t)$ по умолчанию ссылается на $e_{r_z}^n$:

$$PV(t) = \{ \langle p_m, v_{p_m}(t) \rangle \}_1^M,$$

где $v_{p_m}(t)$ — значение p_m -го свойства $e_{r_z}^n$ в момент времени t .

Ориентируясь на наперед заданные индикаторы в виде \overline{PV} , состав $\overline{E}_{r_z}^n(t)$ можно записать как:

$$\begin{aligned} \overline{E}_{r_z}^n(t) = & \left\{ e_{r_z}^n \mid \left(PV(t) \xrightarrow{f_b} \overline{PV}_1 \right) \vee \dots \right. \\ & \left. \dots \vee \left(PV(t) \xrightarrow{f_b} \overline{PV}_l \right) \vee \dots \vee \left(PV(t) \xrightarrow{f_b} \overline{PV}_L \right) \right\}. \end{aligned} \quad (2)$$

Дадим пояснения касательно $PV(t) \xrightarrow{f_b} \overline{PV}_l$. Из постановки задачи вытекает, что между двумя множествами должно быть установлено взаимно-однозначное соответствие. Индекс 'b' в обозначении f_b говорит о том, что f_b — биекция. Согласно определению биекции каждому $\langle p_m, v_{p_m}(t) \rangle$ должен соответствовать $\langle p_m, [\overline{v}_{p_m}^s, \overline{v}_{p_m}^q] \rangle$. Для этого необходимо, чтобы $v_{p_m}(t) \in [\overline{v}_{p_m}^s, \overline{v}_{p_m}^q]$.

Аналогичным образом формализуем второе условие принадлежности $e_{r_z}^n$ к $\overline{E}_{r_z}^n$, также имеющее непосредственное отношение к участию сущности в предопределенных взаимодействиях. Как и в случае со свойствами, сформируем в первую очередь систему множеств \overline{CV} , определяющих индикативные экземпляры взаимодействий:

$$\overline{CV} = \{ \overline{CV}_1, \overline{CV}_2, \dots, \overline{CV}_c, \dots, \overline{CV}_C \},$$

где \overline{CV}_c — множество экземпляров взаимодействий, в которых могут участвовать экземпляры "ролевых" сущностных типов.

В отличие от свойств, которые отличает простой формат представления $\langle \text{Имя свойства} - p_m, \text{Значение свойства} - v_{p_m} \rangle$, для взаимодействий таких форматов может быть предложено несколько. Возьмем за основу очевидный и одновременно универсальный формат, используемый для представления типов, отнесенных к категории "Слабые сущности". При этом будем руководствоваться следующими соображениями.

1. В общем случае предельная кардинальность, которая может быть достигнута во взаимодействии "ролевых" сущностных типов с другими категориями объектов, — M:M —

многие ко многим. "Слабые сущности" — один из двух допустимых типов структур (наряду с "документальными структурами" [25]), который подходит для хранения результатов подобных взаимодействий. В том числе и взаимодействий с кардинальностью 1:1, 1:M, M:1.

2. "Слабые сущности" способны отражать взаимодействия двух и более категорий объектов типов. В демонстрационных целях ограничимся двумя категориями, одна из которых — "ролевой" сущностный тип, другая — тип, с которым взаимодействует ролевой тип.

3. Результатом взаимодействия, кроме факта самого взаимодействия, может стать совокупность значений, приобретаемых какими-то характеристиками, которые, в противовес подлинным свойствам, трактуются как взаимные свойства [23]. Например, претендент на должность должен иметь квалификацию "преподаватель", стаж работы преподавателем не менее 5 лет и опубликовать не менее трех научных работ за последние четыре года.

Таким образом, с учетом сказанного, множество \overline{CV}_c можно определить следующим образом:

$$\overline{CV}_c = \{ \langle We_w, \{ \langle Id_h, \overline{V} \rangle \}_1^H \rangle \}_1^W,$$

где We_w — w-я "слабая сущность"; Id_h — идентификатор h-го объекта, с которым взаимодействует экземпляр ролевого типа; \overline{V} — множество значений характеристик взаимодействий.

В свою очередь, совокупность взаимодействий и результатов этих взаимодействий, в которых сущность состоит в момент времени t , найдется как

$$CV(t) = \{ \langle We_w, \{ \langle Id_h, V \rangle \}_1^H \rangle \}_1^W.$$

Далее, по аналогии с (2), запишем второе условие принадлежности $e_{r_z}^n$ к $\overline{E}_{r_z}^n$:

$$\begin{aligned} \overline{E}_{r_z}^n(t) = & \left\{ e_{r_z}^n \mid \left(CV(t) \xrightarrow{f_b} \overline{CV}_1 \right) \vee \dots \right. \\ & \left. \dots \vee \left(CV(t) \xrightarrow{f_b} \overline{CV}_c \right) \vee \dots \vee \left(CV(t) \xrightarrow{f_b} \overline{CV}_C \right) \right\}. \end{aligned} \quad (3)$$

Выражения (2) и (3) справедливы, если принадлежность $e_{r_z}^n$ к ролевому подмножеству находится на основании либо совпадения ролевых свойств $\left(PV(t) \xrightarrow{f_b} \overline{PV}_l \right)$, либо ролевых от-

ношений $\left(CV(t) \xrightarrow{f_b} \overline{CV}_c \right)$. Но нередко вопрос о $e_{r_z}^n \in \overline{E}_{r_z}^n$ решается с учетом выполнения и первого, и второго условий.

Если бы отсутствовала альтернативность, то выражение для $\overline{E}_{r_z}^n(t)$ имело бы простую форму:

$$\overline{E}_{r_z}^n(t) = \left\{ e_{r_z}^n \mid \left(PV(t) \xrightarrow{f_b} \overline{PV} \right) \wedge \left(CV(t) \xrightarrow{f_b} \overline{CV} \right) \right\}.$$

Альтернативность же предполагает, что индикаторную миссию выполняют наперед заданные пары $(\overline{PV}_l, \overline{CV}_c)_j \in \overline{PV} \times \overline{CV}$. Индекс j указывает на элемент прямого произведения множеств \overline{PV} и \overline{CV} . Воспользуемся порождающей процедурой для нахождения элементов $\overline{E}_{r_z}^n$:

$$\overline{E}_{r_z}^n(t) = \left\{ \begin{array}{l} e_{r_z}^n \mid \left(PV(t) \xrightarrow{f_b} \overline{PV}_l \right) \wedge \left(CV(t) \xrightarrow{f_b} \overline{CV}_c \right) \\ \text{хотя бы для одной } (\overline{PV}_l, \overline{CV}_c)_j \end{array} \right\}.$$

Завершив построение ролевой модели, еще раз обратимся к отношениям, возникающим между элементами множеств, вовлеченных в ролевой контент (рис. 2), и зададимся следующими вопросами: а не чрезмерно ли усложнена такая система? В какой степени она объективна и отражает естественное положение вещей, а в какой субъективна — задумана исключительно для проектирования баз данных, подчиняющихся априори сформулированным принципам их организации?

Если обратить пристальное внимание на любой из распространенных в мире языков человеческого общения, то во всех них обнаружится большое число существительных с "ролевым" лексическим значением. И в этом плане "ролевые" существительные, будучи продуктом человеческого сознания, — такая же объективная реальность, как и сами сущности.

Если взглянуть на систему ролевых отношений (рис. 2) с точки зрения эффективности организации баз данных, то можно обнаружить транзитивную зависимость, "проходящую" через множество ролей, и исключить эту зависимость. Правда, в этом случае пришлось бы полностью отказаться от богатой семантики, сосредоточенной в ролевых экземплярах, которая делает осмысленной последующее использование ролевых подмножеств в приложениях баз данных.

3. Представление ролей на логическом уровне.

Ролевая подтипизация

До сих пор ролевые объекты никоим образом явно не обозначали себя на концептуальном уровне. Роли присутствовали в КМ неявно в виде ссылок на именованные подмножества объектов, которые в состоянии играть ту или иную роль. И этого было вполне достаточно, чтобы представить ролевую составляющую связи между объектами.

Включение ролей в концептуальные модели автоматически влечет за собой решение вопроса о способе их представления в логических моделях. Сделать это можно несколькими способами. Рассмотрим все известные способы, выделив положительные и отрицательные стороны каждого решения.

Но первое, с чем следует определиться и от чего далее будем отталкиваться — каркас будущей логической модели, образуемый типами сущности. Как было отмечено ранее, начальный этап моделирования данных всегда предполагает разбиение множества сущностей Про на непересекающиеся подмножества, именуемые типами. (Чтобы различать другие виды "разбиений", будем именовать подобное разбиение характеристическим, или характеристической подтипизацией (ХП).) Основанием для проведения ХП выступает необходимость в обособлении групп однородных сущностей, отличающихся одинаковыми наборами подлинных свойств. Тем самым достигается неразрезанность таблиц (по одной для каждой группы), в которых предполагается размещать экземпляры этих групп. Заметим, что процедура разбиения носит эвристический характер, и типы сущностей как объекты логических моделей появляются уже на концептуальном уровне. Будем опираться на каркас, образованный типами сущностей, полученными в процессе выполнения ХП, и полагать его отправной точкой для последующего ролевого моделирования.

На основании полученного ранее формального описания роли сформулируем основные требования, которым должны удовлетворять логические структурные подсхемы, инкапсулирующие сущности и роли.

Подсхемы должны:

- обеспечивать смешивание экземпляров различных типов сущностей при реализации ими одинаковых ролей (канонический пример — роль покупателя, которую могут одновременно играть и личности, и организации, если экземпляры личностей и орга-

низаций относятся к разным типам) (требование 1);

- быть в состоянии бесппроблемно (не нарушая принцип независимости программ и данных) выделять в логической и физической моделях ролевые подмножества сущностей, участвующие во взаимодействиях. Более конкретно — запросы к базам данных должны манипулировать исключительно названиями таблиц и не ссылаться на записи таблиц (требование 2).

Если руководствоваться изложенными выше соображениями о назначении и функции роли, то принципиальных решений по "ролевому" логическому моделированию может быть только два (рис. 3). Первое предполагает использование специализированного ролевого типа, содержащего список ролей. Второе — создание типов (по одному типу — "ролевому" сущностному типу — на каждую роль), в которых перечисляются сущности, играющие ту или иную роль. Несмотря на то что последнее решение не создает трудностей в выполнении требования 2, оно не может считаться полностью приемлемым, так как его использование ведет к необоснованному дублированию данных в "ролевых" сущностных типах. (Напомним, одна и та же сущность может играть множество ролей и поэтому будет одновременно присутствовать в качестве экземпляров нескольких типов.) Другой серьезный изъян — вследствие множественности и непостоянного числа ролей размерность модели данных, с одной стороны, будет превышать любые разумные пределы, а с другой, — состав модели будет периодически пересматриваться всякий раз, когда будет появляться очередная роль.

Указанных недостатков лишены две подсистемы, задействующие ролевые типы (рис. 3, б). Правда, и они не обеспечивают выполнение тре-

бования 1. Обратимся к стандартному варианту запроса (см. ниже *Листинг 1*) на выборку сущностей, играющих конкретную роль. Сделаем это на примере типа сущности "Личности (Persons)". Запрос ссылается на ролевой тип "Roles".

Листинг 1

```
SELECT Persons.Surname, Persons.Name, Persons.Patronymic
FROM Roles INNER JOIN Persons_Roles ON Roles.Id_role = Persons_Roles.Id_role LEFT OUTER JOIN Persons ON Persons_Roles.Id_prs = Persons.Id_prc
WHERE (Roles.Id_role = 2)
```

Такой запрос в предложении WHERE всегда будет содержать явный указатель (в нашем примере 2) на конкретную роль (Roles.Id_role = 2). Но при этом следует иметь в виду, что таблица ролей в процессе эксплуатации базы данных заполняется произвольным образом. Поэтому на этапе проектирования логической модели неизвестно, какой идентификатор в базе данных будет присвоен конкретной роли и каким будет сам ролевой список — сколько и каких элементов он будет включать.

Справедливости ради следует заметить, что литерал '2' можно было бы заменить переменной и присваивать ей значение в программном коде, предусмотрев некоторый интерактив, во время которого пользователь мог бы выбрать нужную роль из соответствующего списка. Но подобный выход, что очевидно, явно усложнит работу с приложением. Тем не менее, какой-либо другой альтернативы, кроме как использовать ролевые типы, не просматривается. (По крайней мере, в системе тех канонических понятий, принципов и категорий моделирования, которые сегодня применяются для организации баз данных.) Исходя из этого решение проблем ролевого моделирования будем искать в плоскости трансформации схемы (рис. 3, б), что и будет сделано далее.

Как развитие схемы с ролевым типом на рис. 3, б приведен еще один вариант, в котором ролевые типы непосредственно привязаны к сущностным типам. Несмотря на то что при такой организации также не выполняется требование 1 и частично требование 2 (для его выполнения потребуется продублировать уже экземпляры ролей), примем за основу данную схему, потому что, как показывает практика и что будет продемонстрировано ниже, наблюдаются заметные различия в процедурах, связанных с формированием экземпляров ролей, "играемых" различными сущностными типами.

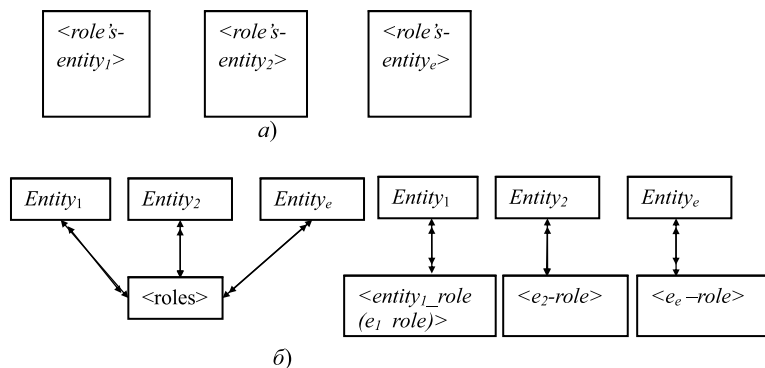


Рис. 3. Способы представления ролей в логических моделях:

а — с "ролевыми" сущностными типами; б — с ролевыми типами

Вернемся к исходному разбиению множества сущностей ПрО, задающему каркас и концептуальной, и логической моделей данных в виде совокупности непересекающихся типов. Не будем останавливаться на возможных вариантах подобных разбиений — их может быть неограниченное число, и это не предмет настоящего исследования. Заметим, что роли выполняют фактически схожую функцию — формируют те же множества сущностей, но только на основании другого признака — участия сущностей в определенных взаимодействиях. Что это дает?

Ролевая подтипизация — это, по сути, один из универсальных адаптационных инструментов моделирования данных, позволяющий получить для любого состояния предметной области ее адекватную модель, в том числе исправить ошибки или неточности, допущенные во время проведения характеристической подтипизации. Например, типы сущностей "Студенты" и "Преподаватели" без труда приводятся к типу "Личности" простой заменой <тип "Личности"> на <роль "Личности">.

4. Абстрактные роли

Как было уже сказано, решение задач ролевого моделирования следует искать в плоскости трансформации схемы (рис. 3, б) с явно выделенными ролевыми типами — по одному ролевому типу на каждый сущностный тип. Обратим внимание на то, что и концептуальный, и логический уровни моделирования не в состоянии оперировать экземплярами *реальных* ролей, которые появляются только по мере заполнения базы данных конкретной предметной области. Следовательно, роли и в КМ, и в ЛМ могут быть только *абстрактными*. Поставим в соответствие, пока только на концептуальном уровне, каждой компоненте отношения по одной абстрактной роли — так, как это было сделано на рис. 1. Заметим, что в этом случае абстрактные роли становятся такими же полноправными объектами КМ, как и другие объекты. Полноправными означает, что их число в КМ — число постоянное, и им всем присвоены уникальные имена.

Следующая задача — каким образом абстрактные роли представить в ЛМ и как связать их с ролевыми типами, в которых будут перечислены реальные роли? Сделать это можно достаточно просто, если разместить список абстрактных ролей в специализированном типе "Abstract_Roles — Абстрактные роли".

Фактически, абстрактные роли — это такие же константы ЛМ, каковыми являются таблицы и отношения: их число известно, и им всем присвоены уникальные имена. Следовательно, не возникает никаких сложностей при обращении ко всем ним из программного кода.

Включение в состав логической модели таких ролевых констант несколько нарушает каноническое представление о видах ее компонентов, но в то же время постоянство состава абстрактных ролей, вытекающее из постоянства отношений логической модели, дает основание для подобного расширения логической модели, не вступая в какие-либо противоречия с принципами организации баз данных. Таким образом, к классическим объектам логических моделей — таблицам и ассоциациям — добавляется конечное множество априори идентифицированных записей таблицы "Абстрактные роли". Связав сущностные таблицы с их ролевыми таблицами и установив связи между ролевой таблицей и таблицей "Абстрактные роли", придем к схеме, показанной на рис. 4.

(Ввиду того, что, во-первых, в ЛМ появляются константы и, во-вторых, не указывается вид ЛМ, термины "таблица" и "тип" используются как синонимы, а структура ЛМ описывается в категориях "таблица", "атрибуты", "отношение" (ассоциация) и запись.)

Удостоверимся в том, что полученная конфигурация действительно позволяет одновременно и объединять (смешивать) сущностные экземпляры, принадлежащие разным типам, когда они играют одинаковые роли, и выделять ролевые подмножества, не прибегая к ссылкам на реальные роли (как это было сделано в листинге 1) при обращениях к данным. Для иллюстрации воспользуемся представлением данных, отражающих факты приобретения то-

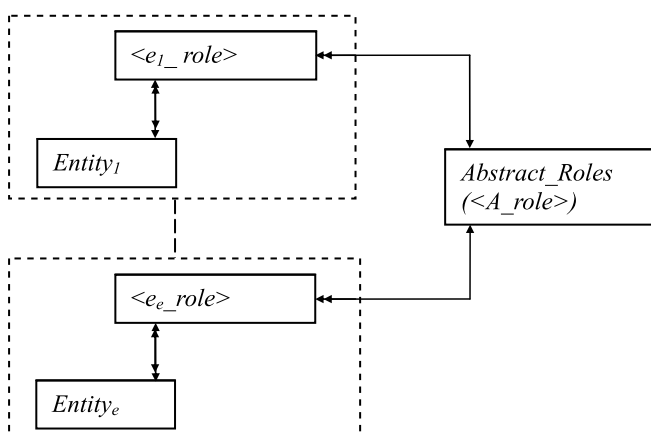


Рис. 4. Abstract_Roles в логической модели

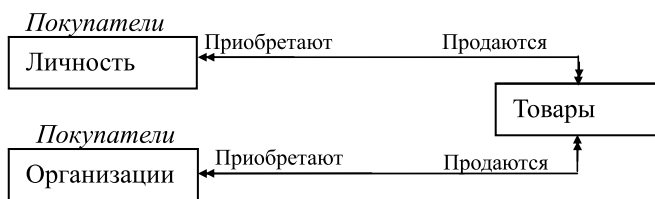


Рис. 5. Концептуальная модель продажи товаров физическим (личности) и юридическим (организации) лицам

варов в некоторой условной предметной области. Концептуальная модель и соответствующая ей логическая модель с фрагментами базы данных изображены на рис. 5 и 6.

Сведения о приобретенных товарах (рис. 6) сосредоточены в таблицах "Purchases_by_Persons" и "Purchases_by_Organizations", отнесенных к категории "слабые сущности — <we>". Ключевые атрибуты помечены символом "*".

Для выборки подмножества личностей, которые в состоянии играть роль преподавателя, воспользуемся запросом (см. ниже Листинг 2), в котором задействована ролевая константа $Id_{ar} = 2$, соответствующая роли "Преподаватель" и размещенная в таблице "Abstract_Roles".

Листинг 2

```

SELECT Person.Sername, Person.Id_prs
FROM Abstract_Roles INNER JOIN
Roles_of_Persons ON Abstract_Roles.Id_ar = Roles_of_Persons.Id_ar INNER JOIN
Persons_Roles ON Roles_of_Persons.Id_RoP = Persons_Roles.Id_RoP INNER JOIN
Person ON Persons_Roles.Id_prs = Person.Id_prs
WHERE (Abstract_Roles.Id_ar = 2)

```

Итоговая выборка будет содержать список лиц (табл. 1), которые являются преподавателями.

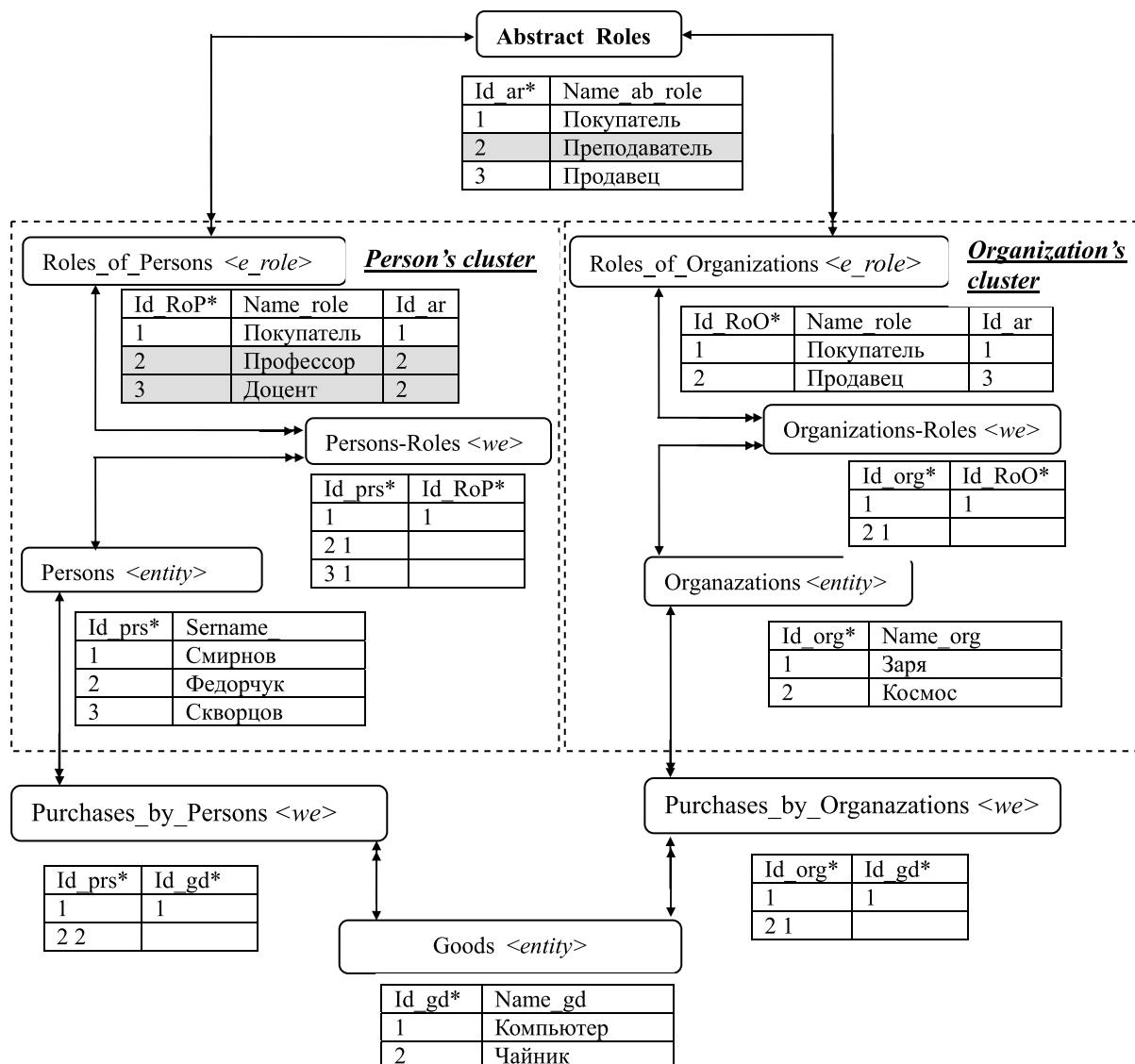


Рис. 6. Фрагмент логической модели с ролевыми объектами и записями базы данных

Таблица 1

Список преподавателей

	Sername	Id_prs
1	Смирнов	1
2	Федорчук	2

Таблица 2

Список покупателей

	Name	Name_gd
1	Смирнов	Компьютер
2	Федорчук	Чайник
3	Заря	Компьютер
4	Космос	Компьютер

Сконфигурировать запрос касательно сущностей, отнесенных к разным типам, но играющих одновременно одну и ту же роль — "покупатель", также не составляет сложностей. Листинг 3 содержит текст SQL-запроса, в результате выполнения которого формируется совмещенный список личностей и организаций, купивших товары (табл. 2).

Листинг 3

```

SELECT Organizatons.Name_org as Name, Goods.
Name_gd
FROM Abstract_Roles INNER JOIN
Roles_of_Organizations ON Abstract_Roles.Id_ar = Roles_
of_Organizations.Id_ar INNER JOIN [Organizations-Roles]
ON Roles_of_Organizations.Id_RoO = [Organizations-
Roles].Id_RpO INNER JOIN Organizatons ON [Organizations-
Roles].Id_org = Organizatons.Id_org INNER JOIN
Purchases_by_Organizatons ON [Organizations-Roles].
Id_org = Purchases_by_Organizatons.Id_org INNER
JOIN Goods ON Purchases_by_Organizatons.Id_gd =
Goods.Id_gd
WHERE (Abstract_Roles.Id_ar = 1)
Union
SELECT Person.Sername as Name, Goods.Name_gd
FROM Persons_Roles INNER JOIN Person ON Persons_
Roles.Id_prs = Person.Id_prs INNER JOIN Abstract_
Roles INNER JOIN Roles_of_Persons ON Abstract_Roles.
Id_ar = Roles_of_Persons.Id_ar ON Persons_Roles.Id_role
= Roles_of_Persons.Id_RoP INNER JOIN Purchases_
by_Persons ON Persons_Roles.Id_prs = Purchases_by_
Persons.Id_prs INNER JOIN Goods ON Purchases_by_
Persons.Id_gd = Goods.Id_gd
WHERE (Abstract_Roles.Id_ar = 1)

```

5. Типы, конкретизирующие роли

До сих пор вопросы, касающиеся состава условных *e_role*-типов и назначения сущно-

стей на роли были вынесены за рамки обсуждения. Между тем *e_role*-типы — это такие же абстракции, как и *A_role*-тип. В "чистом" виде такие типы в ПрО не присутствуют. Их функцию выполняют другие типы.

На примере сущностного типа "Личности", отличающегося развитой и многогранной системой ролевых поведений (о чем свидетельствует богатая лексика, касающаяся обозначений ролей, присутствующая во всех национальных языках), продемонстрируем один из реальных, а не абстрактных *Real_roles*-типов, и рассмотрим круг вопросов, которые возникают в процессе задания ролей и закрепления личностей за ролями.

Типичный *Real_roles*-тип — "Должности". На первый взгляд может показаться, что простого перечисления должностей достаточно, чтобы однозначно конкретизировать роли, играемые личностями в ПрО. Но реальность несколько отличается от подобного упрощенного представления. Нередко роль задается объектами нескольких типов. Например, это может быть пара <"Должности", "Подразделения"> (экземпляр — "Инженер 1-й категории", "отдела IT") или тройка типов <"Должности", "Подразделения", "Совместительство">.

В связи с этим уместно ввести некоторый шаблон ролевого представления, который будет включать обязательный определяющий ролевой тип (<*role*>-объект) и произвольное число уточняющих (необязательных) типов (рис. 7).

На показанном шаблоне (рис. 7) символом 'M' (*Mandatory*) обозначен обязательный тип, а символом 'A' (*Adjunct*) — необязательный (вспомогательный). Экземпляры реальных ролей, представляющие собой некоторое подмножество прямого произведения $M \times A_1 \times \dots \times A_N$, образуют *Person_roles*-объект, который уже не является абстракцией. Чтобы отличать ординарные *e_role*-объекты от реальных ролевых типов, последние на схемах помечаются как *r_role*-типы.

Поскольку экземпляры "*Persons_Roles*" участвуют в последующих взаимодействиях (в частности, с "*Persons*" (рис. 6)), уместно каждому экземпляру "*Persons_Roles*" присвоить уникальный идентификатор и дополнительно классифицировать "*Persons_Roles*" как <*relator*>. Также следует иметь в виду, что в *relator*-ax, которые являются по сути ассоциациями, следует поддерживать уникальность составного внешнего ключа, образованного из атрибутов, ссылающихся на записи M- и A-объектов. На рис. 7 указанная особенность отражена с помощью

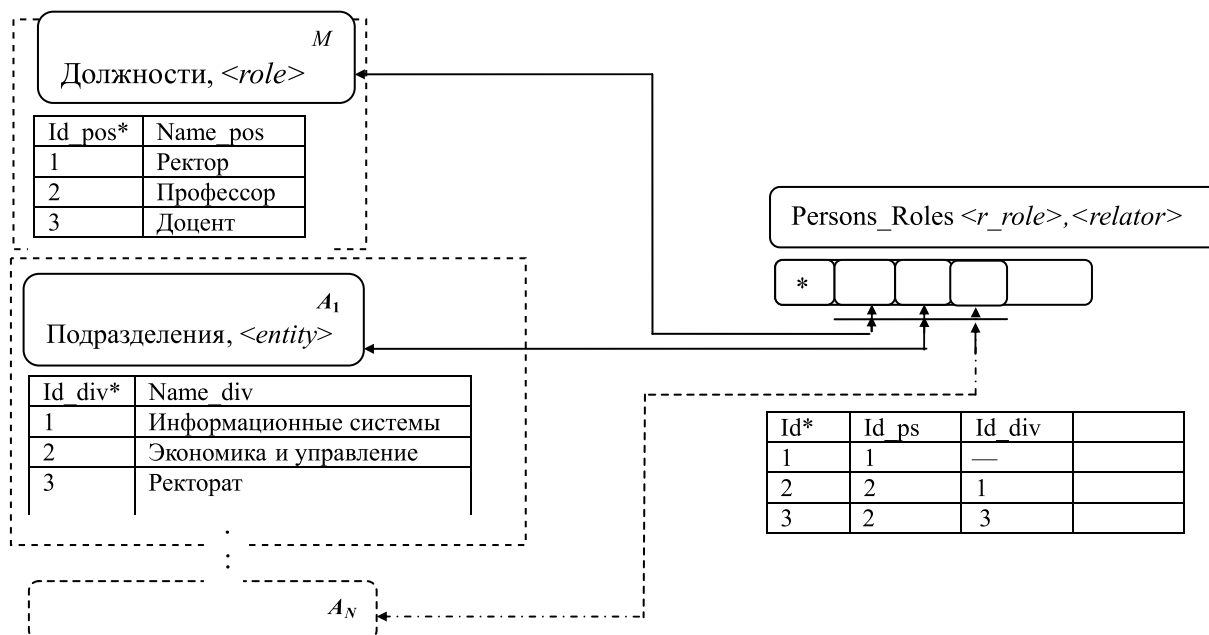


Рис. 7. Шаблон моделирования r_role -типов

сплошной черты, охватывающей все соответствующие атрибуты.

Другая актуальная для ролевого моделирования задача — конфигурирование структурных подсем, записи таблиц которых отражают факты закрепления реальной роли за реальной сущностью. Остановимся на следующей постановке данной задачи: для того чтобы сущность была способна играть ту или иную роль, должны совпасть характеристики этой роли и конкретной сущности. То же самое было ранее представлено в работе в виде формальной модели роли.

Применительно к личностям и их ролям такими характеристиками обычно становятся квалификационные требования, возрастные ограничения и другие аналогичные признаки. В предметных областях нетрудно обнаружить объекты, несущие сведения о квалификациях. На примере моделирования квалификационных требований покажем, что это за объекты, и к каким паттернам моделирования приводит включение типов, содержащих подобные объекты, в состав моделей данных.

6. Назначение на роли и ограничения в использовании ролевых типов

Из представленного ранее определения роли вытекает, что сущность в состоянии играть ту или иную роль, если она (сущность) обладает определенными качествами (свойствами). Например, чтобы стать участником Олимпий-

ских игр, нужно предварительно получить лицензию на участие. Или, к чтению лекций допускаются только лица, имеющие звание доцента или профессора. Следовательно, множество таких характеристик и значений этих характеристик полностью задают конкретную роль. Отсюда следует тот факт, что для того чтобы закрепить за сущностью некоторую роль, достаточно сопоставить свойства роли с аналогичными свойствами сущности. Таким образом, задача назначения сущности на роль распадается на две подзадачи: задачу выявления и моделирования ролевых характеристик и задачу прикрепления роли к сущности.

В предметных областях в качестве носителей "ролевых" характеристик выступают объекты определенной природы, которые, как и роли, также обязаны своим происхождением сущностям (зависимы от них). Способность сущности осуществлять определенные виды деятельности (обобщенные в виде той же должности) оценивается, главным образом, посредством набора квалификационных параметров. Предметные области включают большое число разнообразных групп объектов, специализирующихся на перечислении таких параметров, значения которых обычно ранжированы по возрастанию или убыванию. "Воинские звания", "ученые степени", "ученые звания", "образовательные квалификации" — примеры подобных типов. Чтобы отличать эти типы от типов других категорий, назовем их "ранговыми" типами (<rank>-типы).

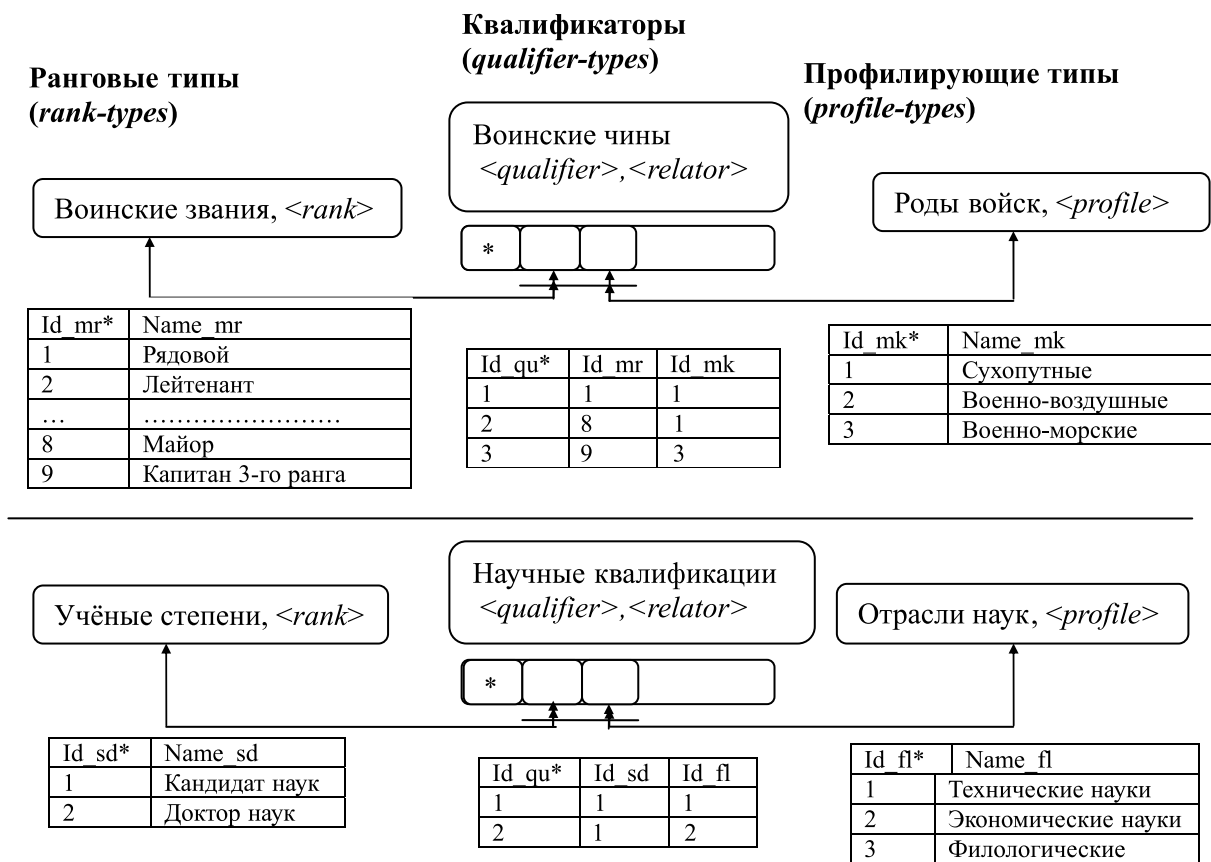


Рис. 8. Квалифицирующий паттерн

Ввиду того что любая профессиональная деятельность осуществляется в специализированных предметных областях, сведения о квалификациях часто дополняются ссылками на сферы профессиональной деятельности, которые, в свою очередь, находят отражение в записях другой категории типов, классифицирующих предметные области или части этих областей. "Роды войск", "Специальности", "Направления подготовки" — примеры таких типов. Поскольку речь в данном случае идет о задании некоего профиля, будем ссылаться на подобные типы как на "профилирующие" типы (<profile>-типы).

Принимая во внимание вышеизложенное, можно прийти к паттерну, моделирующему уровень квалификации сущности (рис. 8).

Как и в случае с *r_role*-типами, квалификационные характеристики также представляются посредством *relator*. Типы, принадлежащие к этой группе, названы "Квалификаторами". Несмотря на то что пример, приведенный на рис.8, не в полной мере показывает специфику предметных областей (в частности, воинские звания следует дополнительно классифицировать на войсковые и корабельные и установить

связь между ними и родами войск), схема содержит все существенные элементы квалификационного паттерна.

Теперь, используя типы, квалифицирующие роли, можно описать характеристики ролей и ролевые свойства сущностей (рис. 9).

Не вызывает затруднений отразить в модели "квалификационные" характеристики сущностей. Достаточно установить отношения между ними и соответствующими квалификаторами.

Кардинальность отношений между сущностным типом и квалификационными типами в общем случае всегда будет М:М. По крайней мере на это указывают многочисленные факты, взятые из практики. Например, личность может одновременно быть и кандидатом

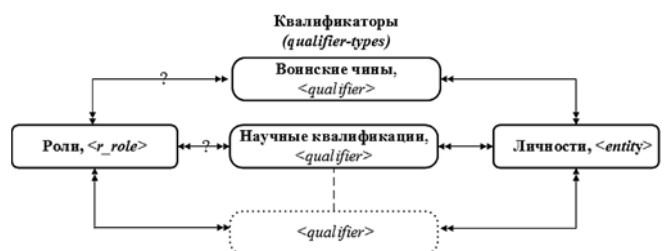


Рис. 9. Соответствия между квалификаторами и сущностями

физико-математических наук, и доктором философии или обладать квалификацией инженера-строителя и магистра экономики.

Отношения же между ролевыми и квалифицирующими типами носят несколько иной характер. Дело в том, что как было указано ранее, роли по своей природе многоальтернативны. Это означает, что роль может задаваться несколькими наборами характеристик. (Каждая характеристика — это пара <характеристика, значение характеристики>.) Кроме того, для конкретизации роли могут привлекаться разные наборы квалифицирующих объектов. Сошлемся для примера на должностные инструкции, содержащие требования к кандидатам на занятие должностей. В них присутствует множество альтернатив, отличающихся как

составом характеристик, принадлежащих разным квалификаторам, так и значениями этих характеристик. Поэтому установления одних только ассоциаций между ролью и квалифицирующими объектами недостаточно. Эти ассоциации не в состоянии аккумулировать все сведения о всех возможных альтернативах. Следовательно, в моделирующем паттерне должен появиться тип — <role_alt>, содержащий список альтернатив для каждой роли. Представленная ниже подсхема (рис. 10) устраняет указанный пробел.

Теперь для принятия решения о назначении сущности на роль достаточно провести сравнение квалификационных характеристик сущности с характеристиками всех ролевых альтернатив, и в случае совпадения по крайней мере с одной из них закрепить роль за сущностью — установить связь между экземпляром сущности и экземпляром роли.

Обобщим результаты последних двух разделов, схематически представив последовательную эволюцию шаблонов ролевого моделирования (рис. 11).

В практической деятельности можно воспользоваться любым из них, в зависимости от того, на какой уровень функциональности ориентирована информационная система: простейший (рис. 11, а) или полный (рис. 11, в). Последний предполагает охват всей палитры ролевых объектов: ранговых, профилирующих, квалификационных и др.

Заключение

Принятая в работе трактовка роли как одного из ключевых признаков, на основании которого отбираются объекты, участвующие в определенных классах взаимодействий (другой признак — фазы — в статье упоминается, но не рассматривается), с одной стороны, приводит к единственно возможному способу логического представления роли, а с другой, обнаруживает две взаимодополняющих проблемы:

- поиск способа объединения экземпляров сущностей, принадлежащих разным классам, в один роле-

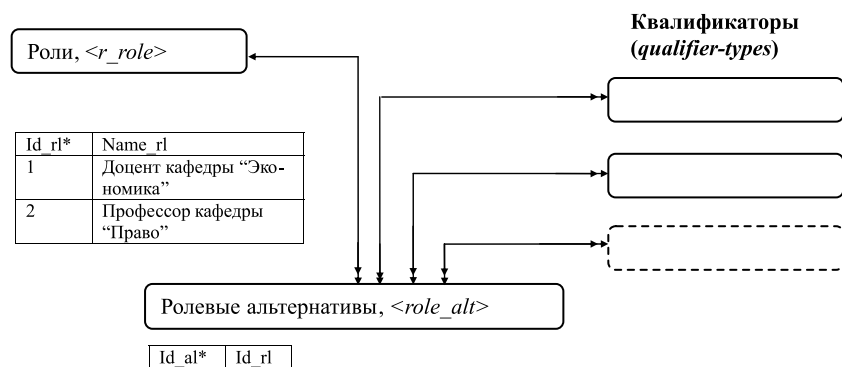


Рис. 10. Шаблон, моделирующий "ролевые" альтернативы

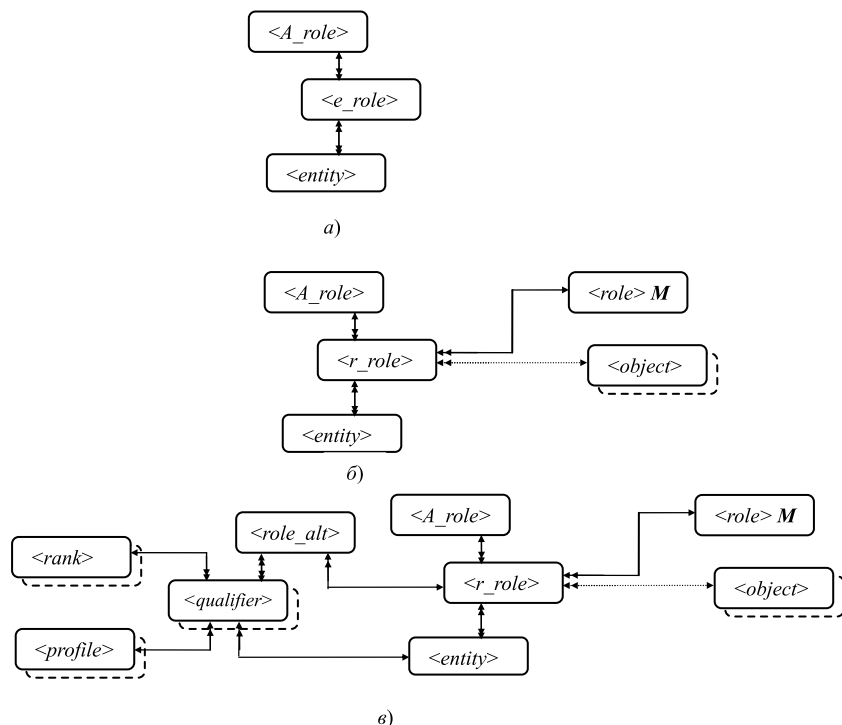


Рис. 11. Эволюция ролевых шаблонов

вой тип, если эти экземпляры в состоянии играть одну и ту же роль. При этом организация логической модели должна исключать появление данных, дублирующих друг друга, и не создавать непреодолимых трудностей доступа к ролевым данным — SQL-запросы не должны содержать явных ссылок на экземпляры ролей;

- неопределенность ролевого прогнозирования, когда требуется отразить в модели экземпляры ролевого типа, которые станут известны только в процессе эксплуатации базы данных.

В работе предлагается и рассматривается способ решения обеих задач, принимающий во внимание два отличительных свойства логической модели: постоянство ее структурной схемы и однозначность идентификации таблиц и ассоциаций, являющихся основными компонентами схемы. Вводится понятие абстрактной роли, и каждой компоненте ассоциации ставится в однозначное соответствие абстрактная роль. Таким образом, список абстрактных ролей становится такой же условной константой логической модели, как и перечень таблиц и ассоциаций. За счет последующей привязки экземпляров реальных ролей к абстрактным достигается и беспроблемное смешивание сущностей (решается первая проблема), и отпадает необходимость в ролевом предвидении.

Предложены моделирующие подсхемы, организованные в виде шаблонов, что делает их пригодными к практическому применению. Шаблоны включают новые категории типов, вовлеченные в орбиту ролевых взаимодействий: *rank-*, *profile-*, *qualifier-*, *role-alt-* и *r-role-* категории.

Полученные результаты показали тесную взаимосвязь между ролевой и фазной подтипизациями, что может составить предмет дальнейших исследований.

Список литературы

1. **Steimann F.** On the representation of roles in object-oriented and conceptual modeling // *Data and knowledge engineering*. 2000. Vol. 35. P. 83–106.
2. **Halpin T., Morgan T.** Information modeling and relational databases. Morgan Kaufmann Publishers, 2008. 970 p.
3. **Буч Г., Рамбо Д., Якобсон И.** Язык UML. Руководство пользователя. М.: ДМК Пресс, 2006. 496 с.
4. **Chen P.** The entity relationship model — toward a unified view of data // *ACM Transaction on Database Systems*. 1976. Vol. 1, N. 1. P. 9–36.
5. **Snoeck M., Dedene G.** Generalization/specialization and role in object oriented conceptual modeling // *Data and knowledge engineering*. 1996. Vol. 19. P. 171–195.
6. **Halpin T. A., Proper H. A.** Subtyping and polymorphism in object-role modeling // *Data and knowledge engineering*. 1995. Vol. 15. P. 251–281.
7. **A. H. M. ter Hofstede and Th. P. van der Weide.** Expressiveness in conceptual data modeling // *Data and knowledge engineering*. 1993. Vol. 10. P. 65–100.
8. **Dahchour M., Pirotte A., Zimanyi E.** A role model and its metaclass implementation // *Information systems*. 2004. Vol. 29. P. 235–270.
9. **Kristensen B. B.** Object-oriented modeling with roles, in: J. Murphy, B. Stone (Eds.), *OOIS'95 // Proceedings of the international conference on object-oriented information systems*, Dublin, Springer. 1996. P. 57–71.
10. **Gottlob G., Schrefl M., Röck B.** Extending Object-Oriented Systems with Roles // *ACM Transactions on Information Systems*. 1996. Vol. 14 (3). P. 269–296.
11. **Elmasri R., Hener A., Weeldreyer J.** The category concept: An extension to the entity-relationship model // *Data and knowledge engineering*. 1985. Vol. 1, N. 1. P. 75–116.
12. **Thalheim B.** Extended entity-relationship model, in: L. Liu, M. T. Ozsu (Eds.), *Encyclopedia of database systems*. Vol. 1. Springer, 2009. P. 1083–1091.
13. **Patig S.** Evolution of entity-relationship modeling // *Data and knowledge engineering*. 2006. Vol. 56. P. 122–138.
14. **Gianni D., Bocciarelli P., D'Ambrogio A.** Referencing capabilities for collaborative engineering of conceptual process modeling with object-role modeling // 2014 IEEE 23-rd international WETICE conference. P. 217–223.
15. **Coulondre S., Libourel T.** Towards a New Role Paradigm for Object-Oriented Modeling // *Proceeding of the OOIS: International conference in object-oriented information systems*. 2002. P. 44–52.
16. **Gogolla M.** UML and OCL in conceptual modeling, in: Embley D. V., Thalheim B. (Eds.) *Handbook of conceptual modeling*, Springer, 2011. P. 85–122.
17. **Keet C. Maria, Fillottrani P. R.** An ontology-driven unifying metamodel of UML Class Diagrams, ERR, and ORM2 // *Data and knowledge engineering*. 2015. Vol. 98. P. 30–53.
18. **Guizzardi G.** Ontological foundations for structural conceptual models. Center for Telematics and Information Technology, University of Twente, The Netherlands, 2005. 441 p.
19. **Carvalho V., Paulo J., Almeida A., Fonseca C. M., Guizzardi G.** Multi-level ontology-based conceptual modeling // *Data and knowledge engineering*. 2017. Vol. 109. P. 3–24.
20. **Родионов А. Н.** Семантическая идентификация, конфигурирование и моделирование типов сущностей в моделях данных // *Вестник НГУ. Серия: Информационные технологии*. 2014. Т. 12, № 1. С. 64–78.
21. **Джексон Г.** Проектирование реляционных баз данных для использования с микроЭВМ. М.: Мир. 1991. 252 с.
22. **Цикритзис Д., Лоховски Ф.** Модели данных. М.: Финансы и статистика, 1985. 168 с.
23. **Cabot J., Raventos R.** Conceptual modeling patterns for roles // *Journal on data semantics V, LNCS 3870*. 2006. P. 158–184.
24. **Parson J., Cole L.** An experimental examination of property precedence in conceptual modeling // *Proceedings of the First Asia-Pacific Conference on Conceptual Modeling (APCCM2004)*. 2004. P. 101–110.
25. **Родионов А. Н.** "Мигрирующие" объекты моделей данных: "слабые сущности" и документы // *Вестник ХГАЭП*. 2011. № 1. С. 40–65.

The Abstract Roles and the Primitives of Role Modeling in the Conceptual, Logical and Physical Data Models System

The role concept is regularly used in the development of conceptual data models. With its help, in particular, it becomes possible to dynamically allocate subsets of entities that are allowed to participate in certain interactions. The embodiment of role behavior at the logical level of modeling can be achieved either by multiple cloning of entity's types—one clone per role, or by means of creating specialized role objects and attaching them to certain types of entities. At the same time, both approaches give rise to concomitant problems, which, as in the first case, lead to the uncontrolled growth of clone objects in logical models, or, as in the second, require dynamic assignment to each role of a constant, which will later place explicitly in queries that manipulate database records. The latter will inevitably lead to a violation of the programs and data independence principle. Proposed and implemented in this paper the concept of abstract roles eliminates these shortcomings of role types and leverage on the information contained in logical models. A unified role modeling pattern with a single abstract role type containing a finite list of abstract roles is developed. The list is constant for the selected logical model configuration. The pattern includes real role types, each of which is tied to its own entity type. Such an organization also addresses the issue of forming a virtual set of entities that play the same role, but belong to different entity types.

Keywords: abstract roles, role modeling, role patterns

DOI: 10.17587/it.25.451-466

References

1. **Steimann F.** On the representation of roles in object-oriented and conceptual modeling, *Data and Knowledge Engineering*, 2000, vol. 35, pp. 83–106.
2. **Halpin T., Morgan T.** Information modeling and relational databases, Morgan Kaufmann Publishers, 2008, 970 p.
3. **Buch G., Rambo D., Jakobson I.** The Unified Modeling Language. User Guide, Moscow, DMK Press, 2006, 496 p. (in Russian)
4. **Chen P.** The entity relationship model — toward a unified view of data, *ACM Transaction on Database Systems*, 1976, vol. 1, no. 1, pp. 9–36.
5. **Snoeck M., Dedene G.** Generalization/specialization and role in object oriented conceptual modeling, *Data and Knowledge Engineering*, 1996, vol. 19, pp. 171–195.
6. **Halpin T. A., Proper H. A.** Subtyping and polymorphism in object-role modeling, *Data and knowledge engineering*, 1995, vol. 15, pp. 251–281.
7. **A. H. M. ter Hofstede and Th. P. van der Weide.** Expressiveness in conceptual data modeling, *Data and Knowledge Engineering*, 1993, vol. 10, pp. 65–100.
8. **Dahchour M., Pirotte A., Zimanyi E.** A role model and its metaclass implementation, *Information Systems*, 2004, vol. 29, pp. 235–270.
9. **Kristensen B. B.** Object-oriented modeling with roles, in: J. Murphy, B. Stone (Eds.), *OOIS'95, Proceedings of the international conference on object-oriented information systems*, Dublin, Springer, 1996, pp. 57–71.
10. **Gottlob G., Schrefl M., Röck B.** Extending Object-Oriented Systems with Roles, *ACM Transactions on Information Systems*, 1996, vol. 14 (3), pp. 269–296.
11. **Elmasri R., Henger A., Weeldreyer J.** The category concept: An extension to the entity- relationship model, *Data and knowledge Engineering*, 1985, v. 1, no. 1, pp. 75–116.
12. **Thalheim B.** Extended entity-relationship model, in: L. Liu, M. T. Oszu (Eds.), *Encyclopedia of database systems*, vol. 1, Springer 2009, pp. 1083–1091.
13. **Patig S.** Evolution of entity-relationship modeling, *Data and knowledge Engineering*, 2006, vol. 56, pp. 122–138.
14. **Gianni D., Bocciarelli P., D'Ambrogio A.** Referencing capabilities for collaborative engineering of conceptual process modeling with object-role modeling, *2014 IEEE 23-rd international WETICE conference*, pp. 217–223.
15. **Coulondre S., Libourel T.** Towards a New Role Paradigm for Object-Oriented Modeling, *Proceeding of the OOIS: International conference in object-oriented information systems*, 2002, pp. 44–52.
16. **Gogolla M.** UML and OCL in conceptual modeling, *Handbook of conceptual modeling*, Springer, 2011, pp. 85–122.
17. **Keet C. Maria, Fillostrani P. R.** An ontology-driven unifying metamodel of UML Class Diagrams, ERR, and ORM2, *Data and knowledge engineering*, 2015, vol. 98, pp. 30–53.
18. **Guizzardi G.** Ontological foundations for structural conceptual models. Center for Telematics and Information Technology, University of Twente, The Netherlands, 2005, 441 p.
19. **Carvalho V., Paulo J., Almeida A., Fonseca C. M., Guizzardi G.** Multi-level ontology-based conceptual modeling, *Data and knowledge Engineering*, 2017, vol. 109, pp. 3–24.
20. **Rodionov A. N.** Semantic Identification, Configuration And Entities Types Modeling For The Data Model Engineering, *Vestnik NGU. Serija: Informacionnye tehnologii*, 2014, vol. 12, no. 1, pp. 64–78 (in Russian).
21. **Dzhekson G.** Relational database design with microcomputer applications, Moscow, Mir, 1991, 252 p. (in Russian).
22. **Cikritzis D., Lohovski F.** Data Models, Moscow, Finansy i statistika, 1985, 168 p. (in Russian).
23. **Cabot J., Raventos R.** Conceptual modeling patterns for roles, *Journal on data semantics V*, LNCS 3870, 2006, pp. 158–184.
24. **Parson J., Cole L.** An experimental examination of property precedence in conceptual modeling, *Proceedings of the First Asia-Pacific Conference on Conceptual Modeling (APCCM2004)*, 2004, pp. 101–110.
25. **Rodionov A. N.** "Migrating" objects of data models: "Weak entities and documents", *Vestnik HGAIeP*, 2011, no. 1, pp. 40–65 (in Russian).

Ю. А. Зак, д-р техн. наук, e-mail: yuriy_zack@hotmail.ru,
Аахен, Германия

Алгоритмы оперативного перераспределения ресурсов в системах сетевого планирования и управления выполнением проектов

Рассматриваются различные стратегии и алгоритмы эффективного распределения ограниченных объемов ресурсов в процессе выполнения заданий проекта, последовательность выполнения которых задана сетевым графиком. На различных этапах разработки определены наиболее важные комплексы работ, требующие увеличения объемов направляемых ресурсов в целях сокращения времени их выполнения. Предложенные в работе методы позволяют сократить время критического пути завершения выполнения как отдельных заданий, так и всего комплекса работ проекта. Предложенные алгоритмы иллюстрируются числовыми примерами.

Ключевые слова: сетевой график, критический путь, алгоритм распределения ресурсов

Введение

Системы СРМ (метод критического пути) и PERT (Program Evaluation and Review Technique — метод оценки и обзора программ) базируются на идее сетевого планирования [1, 8—11], предназначены для оптимизации длительности выполнения сложных проектов за счет эффективных логико-управленческих решений по перераспределению ресурсов для выполнения наиболее важных в данный момент времени заданий, позволяющих, тем самым, управлять временем, необходимым для выполнения заданий.

Представим систему PERT в виде сетевого графика, т. е. графа, вершины которого отображают задания или работы, входящие в состав проекта, а дуги — взаимосвязи этих заданий, т. е. последовательности и очередности их выполнения. Такое представление дает возможность анализа времени, которое требуется для выполнения каждого отдельного задания, расчета минимально возможного и допустимого наиболее позднего времени его начала и завершения, определения минимального необходимого времени завершения проекта, а также перераспределения ресурсов в любой момент времени на выполнение различных заданий для эффективного управления всеми этими показателями.

Разработанные в США в конце 50-х годов методики сетевого планирования позволяют решать определенный круг задач, основными среди которых являются [1—3, 8—11]:

- разработка расписания выполнения всех работ проекта, как с учетом, так и без учета ограниченности ресурсов;

- определение критического пути и резервов времени выполнения работ проекта;
- определение потребности проекта в необходимых ресурсах (финансировании, материалах и оборудовании, составе исполнителей и т. п.);
- анализ возможных рисков и планирование расписания выполнения работ с учетом возможных отклонений от планируемых сроков выполнения отдельных заданий;
- анализ отклонений хода выполнения работ от запланированного и прогнозирование изменений основных параметров проекта.

Если вершины графа определяют задания, входящие в состав проекта, а дуги — их взаимосвязи и последовательности их выполнения, то начало проекта — это вершина без входящих, а окончание — вершина без исходящих дуг. Начало и конец реализации проекта представлены множеством путей, длины которых различаются. Наибольшая длина пути определяет фактическую длительность выполнения всего проекта и называется критическим путем. Критический путь определяет все задания проекта, которые должны быть закончены точно в установленный срок. Знание критических путей в любой точке проекта — необходимое условие успешного выполнения проекта, а также средство выделения приоритетов заданиям, которые требуют особого внимания и определяют в данный момент сроки выполнения проекта. Многие задания, не лежащие на критическом пути, имеют некоторый резерв для начала и завершения времени их выполнения. Резерв времени — это время, на которое можно увеличить продолжитель-

ность (или начало выполнения) данного задания, не изменяя при этом продолжительность критического пути. Если задание находится на критическом пути, то его резерв времени равен нулю.

В большинстве публикаций по анализу методом PERT времени выполнения сложных проектов времена выполнения отдельных заданий рассматривались как установленные экспертами детерминированные величины. Планируемая продолжительность выполнения заданий определялась как некоторое действительное число, вычисляемое по одной из формул:

$$t_i = \frac{1}{5}(3h_i^1 + 2h_i^2), \quad t_i = \frac{1}{6}(h_i^1 + 4\bar{h}_i + h_i^2),$$

где \bar{h}_i — наиболее вероятная оценка, а h_i^1, h_i^2 — соответственно нижняя (оптимистическая) оценка, определяющая время выполнения этого задания в идеальных условиях, и верхняя (пессимистическая) оценка, определяющая максимальную продолжительность с учетом всех возможных срывов.

Анализ сетевых графиков и всех возможных рисков срыва плановых сроков завершения проекта методами fuzzy-арифметики в условиях, когда времена выполнения заданий представлены нечеткими множествами треугольного и трапециевидного вида, рассматривался в работах автора [4—6]. Методам эффективного распределения ресурсов в процессе выполнения работ, обеспечивающих сокращение длины критического пути и времени выполнения проекта, не уделялось значительного внимания в литературе. Автору известна только одна теоретическая публикация, связанная с решением этой проблемы в непрерывном времени методом принципа максимума с ограничением на фазовые координаты [7], которая, к сожалению, получила ограниченное практическое приложение. В данной статье предлагается практическое приближенное решение задачи распределения ограниченных ресурсов в процессе выполнения проекта в целях обеспечения выполнения всех заданий и завершения выполнения проекта в наиболее короткие сроки.

Рассматриваются также стохастические постановки задач по оценке времени выполнения работ, в которых математическое ожидание и дисперсия продолжительности работ рассчитывались по формулам (см., например, [1, 3, 8, 10, 11]):

$$M[t_i] = \frac{1}{5}(3h_i^1 + 2h_i^2), \quad D[t_i] = \left[\frac{h_i^2 - h_i^1}{5} \right]^2.$$

1. Постановка и метод решения задачи

Обозначим:

$\tilde{U} = \{(i, j) \mid i \in \tilde{I}, j \in \tilde{I}, i \neq j\}$ — множество дуг сетевого графика;

$\tilde{A}(i) = \{j \in \tilde{I} \mid (j, i) \in \tilde{U}\}$ (перед "i") — подмножество вершин, являющихся началом дуг, концом которых является вершина i ;

$\tilde{B}(i) = \{j \in \tilde{I} \mid (j, i) \in \tilde{U}\}$ (после "i") — подмножество вершин, являющихся концом дуг, направленных из вершины i .

В условиях, когда в системах СРМ и PERT в управлении разработками использовались числовые детерминированные значения основных показателей, любое i -е задание (задача, событие, вершина графа), $i \in \tilde{I}$, где $\tilde{I} = \{i, j = 1, \dots, n\}$, входящее в состав проекта, характеризуется следующими временными параметрами:

t_i — время, необходимое для выполнения i -го задания;

g_i^1 — наиболее раннее время начала выполнения i -го задания;

g_i^2 — допустимый наиболее поздний срок начала выполнения i -го задания;

w_i^1, w_i^2 — соответственно наиболее раннее и допустимое наиболее позднее время завершения выполнения i -го задания;

r_i — резерв времени завершения выполнения i -го задания;

$\theta(\tau)$ — текущее время выполнения проекта;

D_n — длина критического пути, т. е. продолжительность времени выполнения проекта на начальном этапе решения задачи.

Каждое i -е задание не может быть выполнено до тех пор, пока не завершится выполнение всех заданий, стоящих непосредственно перед ним, т. е. заданий $j \in \tilde{A}(i)$. Следовательно,

$$g_i^1 = \max_{j \in \tilde{A}(i)} (g_j^1 + t_j), \quad w_i^1 = g_i^1 + t_i, \quad i \in \tilde{I}. \quad (1)$$

Расчет значений g_i^1 и w_i^1 ведется от начала сетевого графика к концу. Для введенной для удобства расчетов фиктивной вершины графа $i = 0$, определяющей начало проекта, значение $t_0 = 0$.

Время допустимого наиболее позднего времени начала g_i^2 и завершения выполнения i -го задания w_i^2 (не увеличивая при этом критический путь выполнения проекта) ведется от конечной вершины сетевого графика к началу и рассчитывается по формулам

$$w_i^2 = \max_{j \in \tilde{B}(i)} w_j^2, \quad g_i^2 = w_i^2 - t_i, \quad i \in \tilde{I}. \quad (2)$$

Длина критического пути определяется значением $D_n = w_n^1 = w_n^2$, т. е. временем завершения

выполнения задания, соответствующего последней вершине графа, для которой $\bar{B}(n) = \emptyset$. Возникшие при этом резервы времени выполнения отдельных заданий определяются по формулам

$$r_i = g_i^2 - g_i^1 \text{ или } r_i = w_i^2 - w_i^1, \quad i \in \tilde{I}.$$

В ряде случаев могут быть определены не времена выполнения, а только необходимые объемы работ, связанные с выполнением отдельных заданий, которые обозначим S_i , $i \in \tilde{I} = \{1, 2, \dots, n\}$. Скорость v_i и, следовательно, время выполнения i -го задания t_i зависит от объемов ресурсов различного вида q_i^p , $p = 1, 2, \dots, P$, $i = 1, 2, \dots, n$, выделенных на выполнение данного задания, т. е. $v_i = \varphi_i(q_i^1, \dots, q_i^p, \dots, q_i^P)$, $t_i = \frac{S_i}{v_i}$. Отметим, что на различных этапах выполнения задания вид функции $v_i = \varphi_i(q_i^1, \dots, q_i^p, \dots, q_i^P)$ может быть различным и зависит от стадии выполнения этого задания. В качестве примера приведем несколько видов этих функций:

$$\begin{aligned} v_i^1 &= \sum_{p=1}^P \beta_i^p q_i^p, \quad v_i^2 = \min_{1 \leq p \leq P} \lambda_i^p q_i^p, \\ v_i^3 &= \min \left[\sum_{p \in P_1} \lambda_i^p q_i^p; \sum_{p \in P_2} \rho_i^p q_i^p \right], \\ v_i^4 &= \psi_i \left(\min_{1 \leq p \leq P} \lambda_i^p q_i^p \right). \end{aligned} \quad (3)$$

Здесь β_i^p , λ_i^p , ρ_i^p — весовые коэффициенты, $\psi_i(\cdot)$ — некоторый вид нелинейной зависимости.

В каждый момент времени выполнения проекта τ система принятия решений располагает ограниченным объемом ресурсов каждого вида $\bar{Q}(\tau) = \{Q^1(\tau), \dots, Q^p(\tau), \dots, Q^P(\tau)\}$, которые могут быть распределены для выполнения различного вида заданий таким образом, чтобы

$$\begin{aligned} Q^p(\tau) &\leq \sum_{i \in \tilde{H}(\tau)} q_i^p(\tau), \\ p &= 1, 2, \dots, P, \quad i = 1, 2, \dots, n. \end{aligned} \quad (4)$$

2. Алгоритмы эффективного перераспределения ресурсов

Очень важно определить время выполнения проекта, наиболее целесообразное для перераспределения ограниченных ресурсов. Обозначим $\tilde{K}(\tau)$ — подмножество доступных для

выполнения заданий в момент времени τ , т. е. подмножество вершин графа, не имеющих входящих дуг: $\tilde{K}(\tau) = \{i(\tau) \in \tilde{I} \mid \tilde{A}[i(\tau)] = \emptyset\}$. Ясно, что в момент времени τ ресурсы целесообразно направлять только на выполнение заданий подмножества $\tilde{K}(\tau)$. Здесь $\tilde{A}[i(\tau)]$ — подмножество оставшихся невыполненных заданий непосредственно предшествующих заданию i в момент времени τ .

Сетевой график выполнения всех заданий проекта представим в виде, когда как в начале, так и на любом этапе выполнения алгоритма решения задачи каждой вершине графа поставлено в соответствие не время, а необходимый объем ресурсов для выполнения каждого задания в момент времени τ , который равен $S_i(\tau)$, $i \in \tilde{I}$, $\tau = 1, 2, \dots, T$. Критический путь из любой вершины графа $j \in \{\tilde{I}/n\}$, рассчитанный в момент времени τ , подмножество вершин которого обозначим $L_n(j, \tau)$ и длина которого равна $D_n(j, \tau)$ (здесь $D_n(\tau) = \max_{i \in \tilde{A}(j)} D_n(i, \tau)$), определяет необходимый суммарный объем работ, необходимый в данный момент времени τ для завершения выполнения всех заданий проекта на пути из вершины j в конечную вершину n .

Рассмотрим различные стратегии распределения ресурсов в момент времени τ .

Стратегия 1. Все объемы различного вида ресурсов $Q^p(\tau)$ направляются на выполнение только одного задания $\bar{i}(\tau) \in \tilde{K}(\tau)$, определяющего начальную вершину критического пути ($D_n(\tau)$), рассчитанного в момент времени τ . Пусть оставшийся объем работ, необходимый для завершения выполнения этого задания, равен $S_{\bar{i}}(\tau)$. Вычислим время, необходимое для выполнения этого задания:

$$\begin{aligned} v_{\bar{i}}(\tau) &= \frac{S_{\bar{i}}(\tau)}{\psi_{\bar{i}}(\tau) \{Q^1(\tau), \dots, Q^p(\tau), \dots, Q^P(\tau)\}}; \\ t_{\bar{i}}(\tau) &= \frac{S_{\bar{i}}(\tau)}{v_{\bar{i}}(\tau)}. \end{aligned} \quad (5)$$

Полагаем $\theta(\tau) = \tau := \tau + t_{\bar{i}}(\tau)$.

Стратегия 2. Определим подмножество вершин графа $\tilde{K}(\tau)$, а также длины всех критических путей выполнения проекта $D_n(j, \tau)$, $j \in \tilde{K}(\tau)$, началом которых является каждая из вершин этого подмножества $\tilde{K}(\tau)$. Обозначим объемы работ, необходимые для завершения выполнения этих заданий, $S_j(\tau)$, а число вершин подмножества $\tilde{K}(\tau)$ — $m(\tau)$. Распределение ресурсов в момент времени τ осуществим в соответствии со следующим алгоритмом:

$$\frac{q_1^p(\tau)}{D_n(1, \tau)} = \dots = \frac{q_j^p(\tau)}{D_n(j, \tau)} = \dots = \frac{q_{m(\tau)}^p(\tau)}{D_n(m(\tau), \tau)}; \quad (6)$$

$$\sum_{j=1}^{m(\tau)} q_j^p(\tau) = Q^p(\tau), \quad p = 1, 2, \dots, P.$$

Рассчитываем времена выполнения каждого из заданий $j \in \tilde{K}(\tau)$ по формулам, аналогичным (5):

$$t_j(\tau) = \frac{S_j(\tau)}{v_j(\tau)}, \quad j = 1, 2, \dots, m(\tau), \quad (7)$$

а также самое раннее время выполнения какого-либо задания из этого подмножества

$$\lambda(\tau) = \min \left(\frac{S_1(\tau)}{v_1(\tau)}, \frac{S_2(\tau)}{v_2(\tau)}, \dots, \frac{S_j(\tau)}{v_j(\tau)}, \dots, \frac{S_{m(\tau)}(\tau)}{v_{m(\tau)}(\tau)} \right). \quad (8)$$

Полагаем $\theta(\tau) = \tau := \tau + \lambda(\tau)$.

Стратегия 3. Эта стратегия предусматривает распределение ресурсов на выполнение заданий подмножества $j \in \tilde{K}(\tau)$ в соответствии с алгоритмом решения задачи:

$$\frac{q_1^p(\tau)}{t_1(\tau)} = \dots = \frac{q_j^p(\tau)}{t_j(\tau)} = \dots = \frac{q_{m(\tau)}^p(\tau)}{t_{m(\tau)}(\tau)},$$

$$\sum_{j=1}^{m(\tau)} q_j^p(\tau) = Q^p(\tau), \quad p = 1, 2, \dots, P.$$

Отметим, что при распределении неделимых ресурсов (например, персональных ресурсов) решение описанных выше задач выполняется в целых числах.

Время $\theta(\tau)$ определяется аналогично правилу стратегии 2 по формуле (8).

Стратегия 4. Перераспределение ресурсов выполняем в дискретные моменты времени $\tau = 1, 2, \dots, T$, используя при этом стратегии 1 или 2, либо на каждом шаге или через несколько шагов меняя стратегию.

Алгоритм эффективного распределения ресурсов

Вводим фиктивную вершину сетевого графика $i = 0$. Обозначим $\tilde{I}^1(\tau)$ и $\tilde{I}^2(\tau)$ соответственно подмножество выполненных и подлежащих выполнению в момент времени τ заданий: $\tilde{I}^1(\tau) \cup \tilde{I}^2(\tau) = \tilde{I}$; $\tilde{I}^1(\tau) \cap \tilde{I}^2(\tau) = \emptyset$. В начале процесса положим $\tau = \theta(\tau) = 0$, а также $\tilde{I}^1(\tau) = \emptyset$, $\tilde{I}^2(\tau) = \tilde{I}$. Алгоритм предусматривает выполнение следующих шагов.

Шаг 1. Определяем подмножество вершин графа, не имеющих входящих дуг, — $\tilde{K}(\tau)$.

Определяем и рассчитываем по формулам (1), (2) критические пути с началом в каждой из вершин $j \in \tilde{K}(\tau)$ в конечную вершину графа n , для которой $\bar{B}(n) = \emptyset$. Длины этих путей соответственно равны $D_j(\tau)$. Переходим к шагу 2.

Шаг 2. В момент времени $\tau = \theta(\tau)$, выбирая одну из описанных выше стратегий 1–4, распределяем все объемы имеющихся в распоряжении в данный момент ресурсов $Q^p(\tau)$, $p = 1, \dots, P$, на выполнение заданий подмножества $j \in \tilde{K}(\tau)$. В соответствии с выражениями или алгоритмами (3), (4), (5) определяем по формулам (5) скорости выполнения каждого из этих заданий $t_j(\tau)$, $j \in \tilde{K}(\tau)$. Рассчитываем наиболее раннее время выполнения одного или нескольких из этого подмножества заданий по формуле (8), которое обозначим $\tau_1 = \bar{\theta}(\tau)$. Обозначим подмножество таких заданий $l \in \tilde{K}_1(\tau)$. Переходим к шагу 3.

Шаг 3. Пересчитываем необходимые объемы работ для выполнения заданий подмножества $\tilde{K}_2(\tau) \subset \{\tilde{K}(\tau) / \tilde{K}_1(\tau)\}$ по формулам

$$S_j(\tau_1) = S_j(\tau) - v_j(\tau)t_j(\tau), \quad j \in \tilde{K}_2(\tau).$$

Определяем $\theta(\tau_1) = \theta(\tau) + \tau_1$. Полагаем $\tilde{I}^1(\tau_1) = \tilde{I}^1(\tau) \cup \tilde{K}_1(\tau)$, $\tilde{I}^2(\tau_1) = \{\tilde{I}^2(\tau) / \tilde{K}_1(\tau)\}$. Если $\tilde{I}^1(\tau_1) = \tilde{I}$ и $\tilde{I}^2(\tau_1) = \emptyset$, то алгоритм завершает свою работу, и время выполнения проекта равно $\theta(\tau_1)$. В противном случае, положив $\tau = \theta(\tau) := \theta(\tau_1)$, переходим к выполнению шага 1.

Иллюстративный пример

Граф, отражающий последовательность выполнения заданий, представлен на рис. 1.

Необходимые объемы работ, связанные с выполнением каждого из заданий, сведены в табл. 1.

В системе используется только один вид ресурсов. В каждый момент времени τ объем ресурсов, необходимых для выполнения подмножества

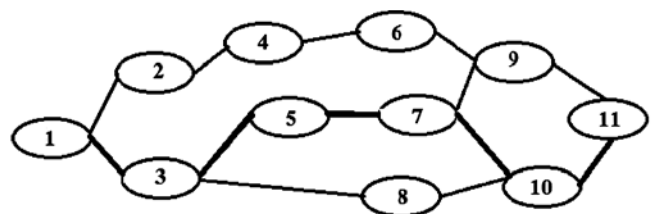


Рис. 1. Граф последовательности выполнения заданий

Исходные данные иллюстративного примера

Номера заданий	1	2	3	4	5	6	7	8	9	10	11
Объемы работ	25	30	15	18	12	28	14	8	30	16	20

Таблица 2

Процесс решения задачи

Номера заданий	2	3	4	5	6	7	8	9	10	11
Объемы работ $T_1 = \theta(\tau_1) = 2,5$	30	15	18	12	28	14	8	30	16	20
Объемы работ $T_2 = \theta(\tau_3) = 6,25$	7,25	—	18	12	28	14	8	30	16	20
Объемы работ $T_3 = \theta(\tau_2) = 7,7$	—	—	18	7,65	28	14	5,1	30	16	20
Объемы работ $T_4 = \theta(\tau_5) = \theta(\tau_8) = 10,25$	—	—	2,55	—	28	14	—	30	16	20
Объемы работ $T_5 = \theta(\tau_4) = 10,675$	—	—	—	—	28	12,3	—	30	16	20
Объемы работ $T_6 = \theta(\tau_7) = 13,75$	—	—	—	—	9,55	—	—	30	16	20
Объемы работ $T_7 = \theta(\tau_6) = 15,35$	—	—	—	—	—	—	—	30	9,6	20
Объемы работ $T_8 = \theta(\tau_{10}) = 17,75$	—	—	—	—	—	—	—	15,6	—	20

жества заданий, равен $Q(\tau) = 10$. Так как речь идет о распределении только персональных ресурсов, объем выделяемых ресурсов на выполнение отдельных заданий может рассматриваться лишь в целых числах. Время выполнения объемов работ в зависимости от выделенных ресурсов определяется линейной зависимостью $S_i(\tau) = v_i(\tau)t_i(\tau)$, где $v_i(\tau) = 1$, и для всех заданий одинаково. Перераспределение ресурсов осуществляется только после завершения выполнения какого-либо задания. Если в некоторый момент времени τ для выполнения i -го задания, оставшийся невыполненным объем которого равен $S_i(\tau)$, выделен объем ресурсов, равный $q_i(\tau)$, то время выполнения этого объема работ равно $t_i = \frac{S_i(\tau)}{q_i(\tau)}$.

Шаг 1. Выделяем все ресурсы на выполнение первого задания. Время выполнения этого задания равно $t_1 = \frac{25}{10} = 2,5$. Текущее время выполнения проекта равно $T_1 = \theta(\tau_1) = \tau_1 = 2,5$.

Шаг 2. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, а. Объемы работ выполнения оставшегося подмножества заданий приведены в первой строке табл. 2, $\tilde{K}(\tau = 2,5) = \{2,3\}$, а критические пути из этих вершин графа представлены соответственно подмножествами вершин $L_2(\tau = 2,5) = \{2,4,6,9,11\}$, длина которого равна $D_2(\tau = 2,5) = 126$, и $L_3(\tau = 2,5) = \{3,5,7,9,11\}$, длина которого равна $D_3(\tau = 2,5) = 91$. Распре-

деление ресурсов на выполнение заданий $\tilde{K}(\tau = 2,5)$ осуществляем в соответствии с соотношением $\frac{91}{126} = \frac{x}{10-x}$. Следовательно, $q_2(\tau = 2,5) = 6$, $q_3(\tau = 2,5) = 4$. Время выполнения третьего задания равно $t_3(\tau = 2,5) = \frac{15}{4} = 3,75$. Текущее время выполнения проекта

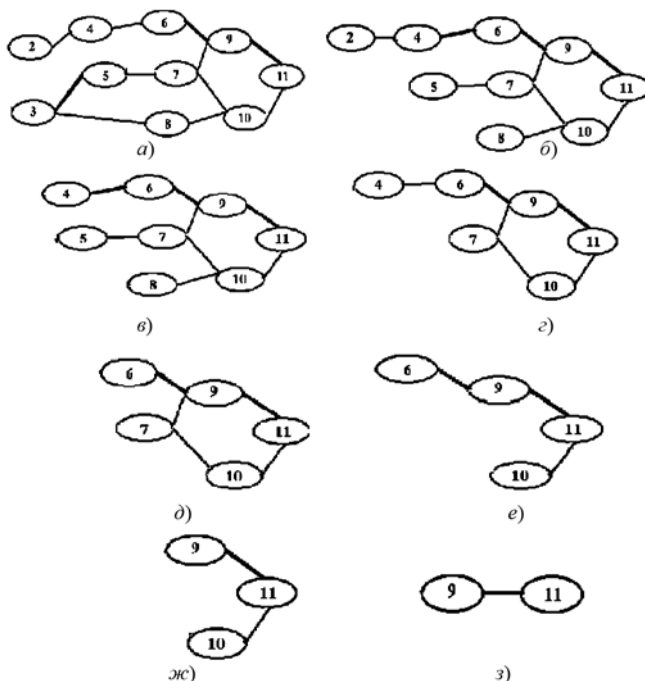


Рис. 2. Граф последовательности выполнения заданий на различных этапах

равно $T_2 = \theta(\tau_3) = 2,5 + 3,75 = 6,25$. Оставшийся объем невыполненного второго задания равен $S_2(\tau = 6,25) = 30 - 6 \cdot 3,75 = 7,25$.

Шаг 3. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, б. Оставшиеся объемы работ выполнения заданий приведены во второй строке табл. 2, $\tilde{K}(\tau = 5) = \{2,5,8\}$, а критические пути из этих вершин графа включают, соответственно, вершины графа $L_2(\tau = 6,25) = \{2,4,6,9,11\}$, длина которого равна $D_2(\tau = 6,25) = 103,25$, графа $L_5(\tau = 5) = \{5,7,9,11\}$, длина которого равна $D_5(\tau = 6,25) = 76$; графа $L_8(\tau = 5) = \{8,10,11\}$, длина которого равна $D_8(\tau = 6,25) = 44$. Распределение ресурсов на выполнение подмножества заданий $\tilde{K}(\tau = 6,25)$ осуществляем следующим образом: $q_3(\tau = 6,25) = 5$, $q_5(\tau = 6,25) = 3$, $q_8(\tau = 6,25) = 2$. Время выполнения второго задания равно $t_2(\tau = 6,25) = \frac{7,25}{5} = 1,45$, $T_3 = \theta(\tau_2) = 6,25 + 1,45 = 7,7$, $S_5(\tau = 7,7) = 12 - 3 \cdot 1,45 = 7,65$, $S_8(\tau = 7,7) = 8 - 2 \cdot 1,45 = 5,1$.

Шаг 4. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, в. Оставшиеся объемы работ выполнения заданий приведены в третьей строке табл. 2, $\tilde{K}(\tau = 7,7) = \{4,5,8\}$, а критические пути из этих вершин графа представлены подмножествами $L_4(\tau = 7,7) = \{4,6,9,11\}$, длина которого равна $D_4(\tau = 7,7) = 96$; $L_5(\tau = 7,7) = \{5,7,9,11\}$, длина которого равна $D_5(\tau = 7,7) = 71,65$; $L_8(\tau = 7,7) = \{8,10,11\}$, длина которого равна $D_8(\tau = 7,7) = 41,6$. Распределение ресурсов на выполнение подмножества заданий $\tilde{K}(\tau = 7,7)$ осуществляем следующим образом: $q_4(\tau = 7,7) = 5$, $q_5(\tau = 7,7) = 3$, $q_8(\tau = 7,7) = 2$. Время выполнения пятого задания равно $t_5(\tau = 7,7) = \frac{7,65}{3} = 2,55$, $T_4 = \theta(\tau_5) = 7,7 + 2,55 = 10,25$, $S_4(\tau = 10,25) = 18 - 5 \cdot 2,55 = 5,25$, $S_8(\tau = 10,25) = 5,1 - 2 \cdot 2,55 = 0$. Следовательно, $\theta(\tau_8) = 10,25$.

Шаг 5. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, г. Оставшиеся объемы работ выполнения заданий приведены в четвертой строке табл. 2, $\tilde{K}(\tau = 10,25) = \{4,7\}$, а критические пути из этих вершин графа представлены подмножествами $L_4(\tau = 10,25) = \{4,6,9,11\}$, длина которого равна $D_4(\tau = 10,25) = 80,55$, и $L_7(\tau = 10,25) = \{7,9,11\}$, длина которого равна $D_7(\tau = 10,25) = 64$.

Распределение ресурсов — $q_4(\tau = 10,25) = 6$, $q_7(\tau = 10,25) = 4$. Время выполнения четвертого задания равно $t_4(\tau = 10,25) = \frac{2,55}{6} = 0,425$, $T_5 = \theta(\tau_4) = 10,25 + 0,425 = 10,675$; $S_7(\tau = 10,25) = 14 - 4 \cdot 0,425 = 12,3$.

Шаг 6. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, д. Оставшиеся объемы работ выполнения заданий приведены в пятой строке табл. 2, $\tilde{K}(\tau = 10,675) = \{6,7\}$, а критические пути из этих вершин графа представлены подмножествами $L_6(\tau = 10,675) = \{6,9,11\}$, длина которого равна $D_6(\tau = 10,675) = 78$, и $L_7(\tau = 10,675) = \{7,9,11\}$, длина которого равна $D_7(\tau = 10,675) = 62,3$.

Распределение ресурсов — $q_6(\tau = 10,675) = 6$, $q_7(\tau = 10,675) = 4$. Время выполнения седьмого задания равно $t_7(\tau = 10,675) = \frac{12,3}{4} = 3,075$, $T_6 = \theta(\tau_7) = 10,675 + 3,075 = 13,75$; $S_6(\tau = 10,675) = 28 - 6 \cdot 3,075 = 9,55$.

Шаг 7. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, е. Оставшиеся объемы работ выполнения заданий приведены в шестой строке табл. 2, $\tilde{K}(\tau = 13,75) = \{6,10\}$, а критические пути из этих вершин графа представлены подмножествами $L_6(\tau = 13,75) = \{6,9,11\}$, длина которого равна $D_6(\tau = 13,75) = 69,55$, и $L_{10}(\tau = 13,75) = \{10,11\}$, длина которого равна $D_{10}(\tau = 13,75) = 36$.

Распределение ресурсов — $q_6(\tau = 13,75) = 6$, $q_{10}(\tau = 13,75) = 4$. Время выполнения шестого задания равно $t_6(\tau = 13,75) = \frac{9,55}{6} = 1,6$, $T_7 = \theta(\tau_6) = 13,75 + 1,6 = 15,35$; $S_{10}(\tau = 13,75) = 16 - 4 \cdot 1,6 = 9,6$.

Шаг 8. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, ж. Оставшиеся объемы работ выполнения заданий приведены в шестой строке табл. 2, $\tilde{K}(\tau = 15,35) = \{9,10\}$, а критические пути из этих вершин графа представлены подмножествами $L_9(\tau = 15,35) = \{9,11\}$, длина которого равна $D_9(\tau = 15,35) = 50$, и $L_{10}(\tau = 15,35) = \{10,11\}$, длина которого равна $D_{10}(\tau = 15,35) = 29,6$.

Распределение ресурсов — $q_9(\tau = 15,35) = 6$, $q_{10}(\tau = 15,35) = 4$. Время выполнения десятого задания равно $t_{10}(\tau = 15,35) = \frac{9,6}{4} = 2,4$, $T_7 = \theta(\tau_{10}) = 15,35 + 2,4 = 17,75$; $S_9(\tau = 15,35) = 30 - 6 \cdot 2,4 = 15,6$.

Таблица 3

Процесс решения задачи

№ шага r	Значения основных параметров в процессе решения					
	Критический путь \bar{L}_r	Длина \bar{L}_r	$j_r \in \{\bar{K}_r\}$	$S(j_r)$	$t(j_r)$	T_r
1	{1, 2, 4, 6, 9, 11}	141	1	25	2,5	2,5
2	{2, 4, 6, 9, 11}	126	2	30	3,0	5,5
3	{3, 5, 6, 9, 11}	105	3	15	1,5	7,0
4	{4, 6, 9, 11}	96	4	18	1,8	8,8
5	{6, 9, 11}	78	6	28	2,8	11,6
6	{5, 7, 9, 11}	76	5	12	1,2	12,8
7	{7, 9, 11}	64	7	14	1,4	14,2
8	{9, 11}	50	9	30	3,0	17,2
9	{8, 10, 11}	44	8	8	0,8	18
10	{10, 11}	36	10	16	1,6	19,6
11	{11}	20	11	20	2,0	21,6

Шаг 9. Граф выполнения заданий на этом этапе выполнения проекта представлен на рис. 2, з. Оставшиеся объемы работ выполнения заданий приведены в шестой строке табл. 2, $\bar{K}(\tau = 17,75) = \{9\}$, а критический путь — $L_9(\tau = 17,75) = \{9,11\}$, длина которого равна $D_9(\tau = 17,75) = 36,6$. В дальнейшем все ресурсы будут направлены на выполнение задания 9, а после его завершения — на выполнение задания 11.

Распределение ресурсов — $q_9(\tau = 17,75) = 10$, $t_9(\tau = 17,75) = \frac{15,6}{10} = 1,56$, $T_8 = \theta(\tau_9) = 17,75 + 1,56 = 19,31$, $q_{11}(\tau = 19,31) = 10$, $t_{11}(\tau = 19,31) = \frac{20}{10} = 2$, $T_9 = \theta(\tau_{11}) = 19,31 + 2 = 20,31$. Следовательно, время завершения выполнения проекта равно 20,31.

В табл. 3 приведен процесс решения задачи алгоритмом, когда весь объем ресурсов в каждый момент времени τ направляется только на выполнение задания, стоящего первым в критическом пути выполнения проекта, и перераспределение ресурсов осуществляется после выполнения этого задания. Время выполнения проекта в этом случае $T_r = 21,6 > 20,31$, т. е. несколько больше полученного описанным выше алгоритмом.

Апробация предложенных методов

Предложенные методы были апробированы при разработке программной системы имитационного моделирования работы предприятий производства электронной аппаратуры. Программная система предусматривала создание следующих подсистем, каждая из которых состояла из множества программных модулей и

разработка которых могла вестись в ряде случаев независимо друг от друга:

- подсистема реализации алгоритмов расчетов динамики и основных параметров производственного процесса;
- банки данных структуры и основных параметров моделируемых производств, а также хранения результатов выполненных вычислений;
- графический интерфейс ввода данных, выбора структурных единиц моделируемых производств, варьирования параметров управляющих воздействий и алгоритмов принятия решений в различных ситуациях;
- графический интерфейс формирования требований выбора объемов и представления формы вывода результатов выполненных расчетов;
- создание различных форм представления входной и выходной информации в удобном для пользователя графическом, табличном виде или в форме анимации;
- создание интерфейса системной увязки программных модулей и подсистем;
- разработка тестов и тестирование программных модулей, подсистем и всего программного комплекса.

Разработка велась группой программистов и системных аналитиков, включающей на различных этапах выполнения проекта 5...7 человек.

Использование первого алгоритма распределения ресурсов, приведенного в иллюстративном примере, позволило сократить планируемые сроки выполнения проекта и завершить разработку проекта в установленные заказчиком сроки.

Заключение

В условиях, когда заданы объемы работ, необходимые для выполнения каждого из заданий проекта, последовательность выполнения которых задана сетевым графиком, а также функциональные зависимости скорости выполнения этих заданий от объемов выделенных ресурсов, эффективное распределение ограниченных ресурсов в течение всего времени выполнения проекта позволяет существенно сократить сроки выполнения как отдельных заданий, так и всего комплекса работ проекта.

Определены наиболее важные комплексы работ, требующие увеличения объемов направляемых ресурсов в целях сокращения времени их выполнения, необходимые моменты времени внесения корректировок в процесс разработки, а также различные стратегии распределения

ограниченных ресурсов и алгоритмы управления процессом выполнения проекта. Предложенные в работе методы, направленные на сокращение времени критического пути выполнения проекта, иллюстрируются числовым примером.

На основе предлагаемых подходов распределения ресурсов для различного вида конкретных функций зависимости скорости выполнения заданий от объемов выделенных ресурсов (которые существенно различаются для реализации проектов в области строительно-монтажных работ, проектирования сооружений, разработки программных комплексов, организации перевозок, сервисного обслуживания и т. п.) могут быть разработаны конкретные алгоритмы и программные комплексы управления проектами различного назначения, которые должны позволить сократить сроки и требуемые объемы ресурсов при выполнении проектов.

Список литературы

1. Матюшок В. М., Бурчакова М. А., Лазанюк И. В., Матюшок С. В., Смаржевский И. А., Сорокин Л. В., Якубова Т. Н. Управление проектами. М.: РУДН, 2010. 553 с.
2. Наумов А. А., Баженов Р. И. О проблемах классических показателей эффективности инвестиционных проектов // Современные научные исследования и инновации. 2014. № 11-2 (43). С. 181—187.
3. Олейникова С. А. Модификация метода PERT решения задач сетевого планирования и управления // Системы управления и информационные технологии. 2008. № 4 (34). С. 42—45.
4. Зак Ю. А. Принятие эффективных решений в экономике и менеджменте в условиях наличия нечисловой информации и размытых данных. М.: Экономика, 2018, 245 с.
5. Зак Ю. А. Принятие решений в условиях размытых и нечетких данных. М.: URSS, 2013. 352 с.
6. Зак Ю. А. Определение экономических параметров выполнения проектов в условиях нечетких данных // Научное обозрение: экономика и управление. 2012. № 4. С. 138—146.
7. Беликов С. А., Зак Ю. А. Некоторые задачи оптимального распределения ограниченных ресурсов при выполнении сетевого графика взаимосвязанных работ // Кибернетика. 1985. № 2. С. 59—67.
8. Hajdu M., Bokor O. The Effects of Different Activity Distributions on Project Duration in PERT Networks // Procedia — Social and Behavioral Sciences. 2014. N. 119. С. 766—775.
9. Baker Sunny G., Michael Campbell, Kim Baker. The Complete Idiot's Guide to Project Management. Alpha Books, 2003.
10. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. John Wiley & Sons, 2003.
11. Punmia B. C., Khandelwal K. Project Planning and Control P. E. R. T. and C. P. M.: For Degree Classes. Laxmi Publications, 2006.

Yu. A. Zack, D. Sc., Aachen, Germany, e-mail: yuriy_zack@hotmail.com

Algorithms for the Operational Redistribution of Resources in Network Planning and Project Management Systems

Various strategies and algorithms for the effective allocation of limited volumes of resources in the process of fulfilling project tasks, the sequence of which is set by the network, are considered. At various stages of development, the most important work packages have been identified that require an increase in the volumes of resources allocated with the aim of reducing their execution time. The methods proposed in the work allow to reduce the time of the critical path to complete the implementation of both individual tasks and the whole complex of project activities. The proposed algorithms are illustrated by numerical examples.

Keywords: network, critical path, resource allocation algorithm

DOI:10.17587/it.25.467-474

References

1. Matyushok V. M., Burchakova M. A., Lasanyuk I. V., Matyushok S. V., Smarghevsky I. A., Sorokin L. V., Yakubova T. N. Project Management, Moscow, RUDN, 2010, 553 p. (in Russian).
2. Naumov A. A., Baghenov R. I. On the problems of classical indicators of the effectiveness of investment projects, *Sovremennije nauchnije issledovaniya i innovazii*, 2014, no. 11-2 (43), pp. 181—187 (in Russian).
3. Olejnikova S. A. Modification of the PERT method for solving problems of network planning and control, *Sistemi upravlenija i informazionnija tehnologij*, 2008, no. 4 (34), pp. 42—45 (in Russian).
4. Zack Yu. A. Making effective decisions in economics and management in the presence of non-numeric information and vague data, Moscow, *Ekonomika*, 2018, 245 p. (in Russian).
5. Zack Yu. A. Decision Making under Blurred and Fuzzy Data, Moscow, UR SS, 2013, 352 p. (in Russian).
6. Zack Yu. A. Determination of the economic parameters of the implementation of projects in terms of fuzzy data, *Nauchnoje obosrenije: Ekonomika i Upravlenije*, 2012, no. 4, pp. 138—146 (in Russian).
7. Belikov S. A., Zack Yu. A. Some problems of optimal distribution of limited resources when performing the network graph of interrelated works, *Kibernetika*, 1985, no. 2, pp. 59—67 (in Russian).
8. Hajdu M., Bokor O. The Effects of Different Activity Distributions on Project Duration in PERT Networks, *Procedia — Social and Behavioral Sciences*, 2014, no. 119, pp. 766—775.
9. Baker Sunny G., Michael Campbell, Kim Baker. The Complete Idiot's Guide to Project Management, Alpha Books, 2003.
10. Kerzner H. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, John Wiley & Sons, 2003.
11. Punmia B. C., Khandelwal K. Project Planning and Control P. E. R. T. and C. P. M.: For Degree Classes, Laxmi Publications, 2006.

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.932.2, 681.516.7.015.2

DOI: 10.17587/it.25.475-481

Ш. С. Фахми^{1,2}, д-р техн. наук, проф., e-mail: Shakeebf@mail.ru,

Н. Г. Рыжов¹, канд. техн. наук, доц., e-mail: ngryzhov@etu.ru,

Х. А. Хасан¹, аспирант, e-mail: midocom@mail.ru,

Е. С. Калинина³, канд. техн. наук, доц., e-mail: elenakali21@yandex.ru,

¹ Санкт-Петербургский государственный электротехнический университет "ЛЭТИ" им. В. И. Ульянова (Ленина),

² Институт проблем транспорта им. Н. С. Соломенко Российской академии наук

³ Санкт-Петербургский университет государственной противопожарной службы РФ

Адаптивный алгоритм кодирования и декодирования изображений в транспортных системах наблюдения

Известно, что стандарты обработки изображений (JPEG) и видео (MPEG), основанные на применении спектрального косинусного преобразования, не удовлетворяют растущим требованиям к качеству передачи видеoinформации, полученной, в частности, из камер наблюдения на транспорте. Причиной этого являются большие объемы визуальных данных и тот факт, что оптимизацию видеокodeка надо делать (как минимум) по трем параметрам: по битовой скорости, по искажению и по вычислительной сложности. Поэтому возникает задача разработки новых алгоритмов сжатия изображений, наилучшим образом согласованных с имеющимися стандартами. Один из алгоритмов совершенствования процесса кодирования и декодирования рассмотрен в настоящей работе. Он отличается от существующих тем, что в нем размеры блоков, подвергающихся косинусному преобразованию, могут быть произвольными и выбираются с учетом специфики транспортных сюжетов.

Ключевые слова: кодирование изображений, косинусное преобразование, адаптивность, транспорт, скорость передачи, сложность

Введение

Интеллектуальные транспортные системы и транспортная инфраструктура имеют огромное количество визуальных данных, получаемых от различных камер наблюдений. Поэтому важной задачей является уменьшение объема видеoinформации при передаче ее в реальном времени при сохранении требуемого качества [1, 2].

Проблема обработки видеoinформации на транспорте, которой десятилетиями занимается большое число организаций и специалистов, имеет множество решений в связи с разнообразием критериев, часто не формализованных, качества априорной информации о сигналах (статистики наблюдаемых сюжетов) и имеющихся ограничений [3]. При этом часто остается открытым вопрос об оптимальности того или иного метода сжатия изображений для дальнейшей передачи и восстановления в реальном времени [4, 5]. Известные методы сжатия изображений чаще всего используют

критерии допустимой ошибки восстановления для достижения требуемого сжатия или максимального приближения к энтальпии источника видеoinформации [6, 7].

Видеонаблюдение на транспорте занимает особое место ввиду существования комплекса специфических требований к особенностям эксплуатации, техническим характеристикам устройств кодирования и декодирования. В первую очередь видеoinформация должна быть доставлена в центры управления в реальном времени, т. е. за нужный интервал, необходимый для принятия решения и обеспечения безопасности.

Специфика заключается еще и в том, что изображения, полученные от камер наблюдения, характеризуются четкой выделенной доминантой (объекта наблюдения), обусловленной априорной информацией о постоянном и неизменном фоне. Другими словами, в условиях интенсивного движения значимость обнаружения и слежения за объектом, в данном случае транспортным средством (ТС), становится кри-

тической, в то время как при низкой интенсивности движения ТС практически отсутствуют и, тем самым, энтропия источника информации равна нулю (наблюдается только фон).

1. Постановка задачи эффективной передачи изображений

Появление в последнее время технологии "система на кристалле" и современных САПР на базе реконфигурируемых систем, подлежащих репрограммированию, привело к формализованному учету сложности устройств кодирования и декодирования изображений. Формализация, стимулированная созданием и развитием новой элементной базы вычислительной техники в виде сложнофункциональных блоков в составе видеосистем на кристалле [8], и учет сложности на этапе синтеза требуют пересмотра понятия "эффективное" кодирование. В круг величин, связанных с эффективностью кодирования нестационарного источника изображений, входят не только точность восстановления (среднеквадратическое отклонение, СКО) и скорость создания кода (число битов на пиксель), но и вычислительная сложность. Эти три величины взаимосвязаны, и выбор устройств кодирования и декодирования необходимо осуществлять по этим трем показателям качества: точности, скорости и сложности [9].

Оценка взаимосвязи информационных показателей качества систем передачи видеoinформации осуществляется на основе предложенного в работе [10] показателя эффективности P , включающего взвешенную сумму $\{P_i\}$ частных информационных показателей качества кодирующих устройств, которые связаны с ошибкой передачи ε :

- степени искажения исходной информации ΔI ;
- скорости передачи R ;
- сложности W_d декодера;
- сложности W_k кодера.

Вектор весовых коэффициентов $\{c_i\}$ при этих показателях является "вектором концепции системы" [11]. Задача оптимизации при синтезе видеосистем на кристалле в целом сводится к поиску оптимальных значений информационных показателей качества при заданных соответствующих весовых коэффициентах, обеспечивающих минимум целевой функции [10]:

$$P = c_0 \Delta I(\varepsilon) + c_1 \Delta R(\varepsilon) + c_2 \Delta W_k(\varepsilon) + c_3 \Delta W_d(\varepsilon) \rightarrow \min.$$

Заметим, что определение множества решений с учетом ограничений для данной функции является труднейшим этапом всего процесса оптимизации при проектировании видеосистем.

В данной работе решаются следующие задачи:

- разработка адаптивного дискретного косинусного преобразования (АДКП) изображений;
- определение области решений, приводящих к минимуму эффективного показателя качества (P);
- сравнение результатов исследований с известными методами кодирования и декодирования изображений для различных классов изображений, полученных в результате наблюдения на транспорте.

Следует отметить, что сжатие изображений заключается в удалении избыточности на изображениях нестационарного источника, и этот процесс называется кодированием источника. Следующим этапом является энтропийное кодирование с последующей передачей по каналу — кодирование канала. Если первый кодер имеет цель — минимизацию скорости передачи, выраженную числом битов на пиксель, то второй кодер должен передать сжатую видеoinформацию с учетом пропускной способности канала связи при максимальном сохранении уже сжатой информации.

В данной статье рассматриваются алгоритмы кодирования нестационарного источника изображений в транспортных видеoinформационных системах наблюдения и контроля. Анализ результатов сжатия и восстановления изображений осуществляется в сравнении с другими методами преобразования, такими как: Карунена—Лоева (ПКЛ) [12], JPEG-DCT-2D (DCT) [13] и дискретное вейвлет-преобразование (ДВП) [14].

2. Описание адаптивного метода кодирования

Тенденция в проектировании видеосистем направлена на использование технологии "система на кристалле" [15], где имеются: процессор общего назначения, интегрированный в виде одного чипа с пользовательскими аппаратными ускорителями, и программируемая логика для аппаратной реализации сложных функций.

В настоящее время современные программируемые схемы (ПЛИС) обеспечивают возможность построения новых видеосистем на кристалле, в состав которых входит фотопри-

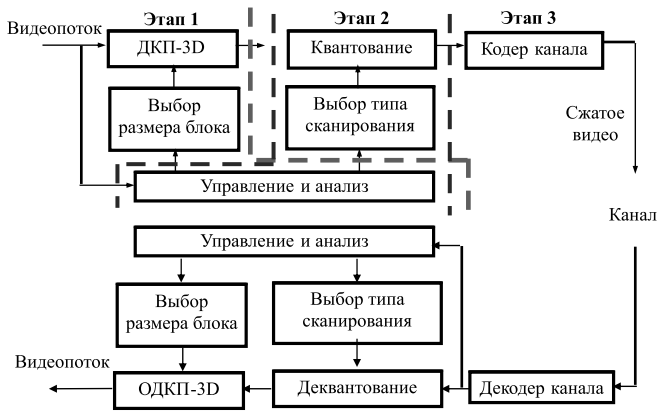


Рис. 1. Структурная схема адаптивного косинусного преобразования

емник. Главные поставщики ПЛИС — это фирмы Intel (в прошлом Altera) и Xilinx, и они сегодня выпускают многопроцессорные реконфигурируемые системы на кристалле, позволяющие реализовать более сложные алгоритмы обработки больших объемов визуальных данных. Наличие в составе системы фотоприемника делает возможным реализацию дополнительных операций на этапе кодирования, в частности, адаптацию и учет статистических характеристик изображений.

Структурная схема кодера на основе АДКП показана на рис. 1. Кодер включает следующие основные блоки: блок управления и анализа статистических характеристик изображений (корреляции, дисперсии (D), коэффициента контраста и т. д.); блок АДКП-2D; блок квантования и блок кодера канала для передачи по каналу связи. Декодер содержит те же блоки, только в обратном порядке.

Процесс кодирования изображений включает три основных этапа.

Этап 1. Предварительный анализ последовательности кадров. Первое изображение последовательности разбивается на равные не пересекающиеся полигоны в зависимости от заданного порога яркости для определения степени контраста. Затем процедура разбиения рекурсивно повторяется для каждого полигона до тех пор, пока его размер не станет равным четырем элементам исходного изображения [15]. В результате такого разбиения получим неравномерную рекурсивную полигональную сетку (рис. 2, см. вторую сторону обложки). Далее каждый полигон данной сетки подвергается дискретному косинусному преобразованию (ДКП).

Важно отметить, что при высоких значениях коэффициента корреляции в пределах по-

лигона (0,7...1) полигон не подвергается ДКП, и всем пикселям на этапе восстановления изображения присваивается среднее значение яркости по полигону (полигон далее по тексту будем называть блоком в соответствии с общепринятой терминологией).

Таким образом, в результате предварительного анализа формируется неравномерная полигональная сетка блоков ДКП. Далее каждую ячейку этой сетки будем называть блоком ДКП (рис. 2, см. вторую сторону обложки):

- блоки 4×4 — области изображения с высоким контрастом (границы объектов), определяются максимальным значением порога по яркости ρ_{max} ;
 - блоки 8×8 и 16×16 — области средней степени контраста изображений, определяются порогами ρ_{ave} ;
 - блоки 32×32 и выше — с наибольшей степенью корреляции значений яркостей и минимумом контраста, определяются порогами ρ_{min} .
- Легко заметить следующие особенности изображений в процессе наблюдения:
- отсутствие транспортных средств;
 - малая интенсивность движения (на кадре только одна машина);
 - высокая интенсивность и наличие потока движения машин (рис. 2, см. вторую сторону обложки).

Использование этих особенностей транспортных изображений в процессе наблюдения за дорогами позволяет обоснованно избегать ненужных вычислений и тем самым существенно сократить вычислительную сложность процесса обработки изображений.

Каждый блок исходного изображения, представляющего собой двумерную матрицу размером $N \times N$, подвергается ДКП по следующей формуле:

$$S(i, j) = \frac{1}{\sqrt{2N}} \alpha(i) \alpha(j) \times \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} M(x, y) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right),$$

где $M(x, y)$ — входная матрица блока ДКП размером $N \times N$; (x, y) — координаты пикселей матрицы и (i, j) — координаты коэффициентов ДКП (т. е. i — номер строки, j — номер столбца), а α вычисляется по следующей формуле:

$$\alpha(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{при } u = 0, \\ 1 & \text{при } u > 0. \end{cases}$$

Обратное косинусное преобразование выполняется по следующей формуле:

$$M(x, y) = \frac{1}{\sqrt{2N}} \times \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \alpha(i)\alpha(j) \cos\left(\frac{(2x+1)i\pi}{2N}\right) \cos\left(\frac{(2y+1)j\pi}{2N}\right).$$

Предложенное АДКП представляет собой ортогональное преобразование яркостей пикселей в целях декорреляции и перевода сигнала изображений из пространственного представления в спектрально-частотное представление. Отличие предложенного алгоритма от известных [16–18] заключается в возможности выбора любого размера блоков ДКП вплоть до размера кадра в зависимости от значений контраста и степени корреляции яркости соседних пикселей.

Другими словами, при разработке кодера на основе предложенного алгоритма АДКП были учтены следующие основные идеи, позволяющие отличить его от всех известных:

- учет специфики изображений транспортной видеосистемы наблюдения;
- применение любого размера блоков для ДКП по заданным порогам корреляции и контраста;
- автоматическое переключение режимов кодера в зависимости от текущего транспортного сюжета.

3. Специфика транспортных сюжетов

Исследование огромного числа изображений (более 200 кадров) транспортной специфики с предложенной в работе [19] мерой неопределенности (Un) и мерой широкомасштабности (Bw) позволяло разделить их на 9 классов (рис. 3, см. вторую сторону обложки).

Легко заметить, что классы: 1) ($Bw_{II}-Un_3$) — изображения с малой широкополосностью и большой неопределенностью, 2) ($Bw_{II}-Un_2$) — изображения со средней широкополосностью и средней неопределенностью и 3) ($Bw_{II}-Un_1$) — изображения со средней широкополосностью и малой неопределенностью имеют относительно большой размер площади фона (небо, дорога, лес), которые являются однородными областями, и они формируются блоками большего размера, следовательно, не подвергаются ДКП (рис. 2, см. вторую сторону обложки), а при восстановлении пикселям данных областей присваивают среднее значение яркости. Это обстоятельство приводит к увеличению

коэффициента сжатия в 2...3 раза по сравнению с другими методами.

Размеры блока ДКП. Для определения степени корреляции областей изображений были использованы формулы на основе метода наименьших квадратов (МНК). Если представить яркости пикселей в виде высот по координате z в пространстве (рис. 4), то задача определения степени корреляции сводится к следующей: найти плоскость, проходящую через множество точек в пространстве при условии, что сумма квадратов отклонений эмпирических значений яркостей пикселей от нее должна быть минимальной.

Таким образом, для нахождения уравнения плоскости (z) тренда, т. е. тенденции изменения значений яркостей пикселей, имеем (рис. 4):

$$F = \sum_{k=1}^{N \times N} \varepsilon_k \rightarrow \min ,$$

где ε — отклонение по яркости пикселей в пределах блока от плоскости тренда (z). Для заданных значений x_i , y_i и z_i блока размером $n \times n$ плоскость тренда находится по следующим формулам:

$$\begin{aligned} a \sum_{i=1}^n x_i + b \sum_{i=1}^n y_i + nc &= \sum_{i=1}^n z_i; \\ a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i y_i + c \sum_{i=1}^n x_i &= \sum_{i=1}^n x_i z_i; \\ a \sum_{i=1}^n x_i y_i + b \sum_{i=1}^n y_i^2 + c \sum_{i=1}^n y_i &= \sum_{i=1}^n y_i z_i; \\ D &= \frac{\sum_{i=1}^N \sum_{j=1}^N (v_{original\ image} - v_z)^2}{NM}, \end{aligned}$$

где $v_{original\ image}$ — исходная яркость пикселя, v_z — значение яркости на плоскости (z).

Эман 2. Адаптивное квантование и сканирование. Для адаптивного квантования полученных матриц коэффициентов ДКП используются

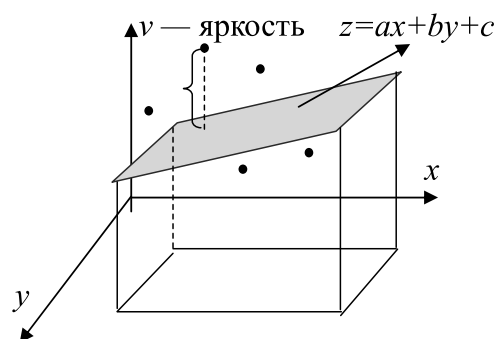


Рис. 4. Оценка степени корреляции области МНК

различные способы сканирования в зависимости от расположения ненулевых коэффициентов. Процесс квантования осуществляется делением на заданный коэффициент квантования $QC_{ij} = 1 + (1 + i + j)R$, где i и j — координаты отсчета внутри матрицы; R — параметр квантования, задаваемый пользователем [9].

Этап 3. Кодирование канала. Квантованные коэффициенты с применением двумерного ДКП (2D-ДКП) считаются и кодируются методом Хаффмана с последующей передачей по каналу [20].

Работа декодера. Процесс декодирования аналогичен процессу кодирования, разница в том, что этапы выполняются в обратном порядке (см. рис. 1).

4. Критерий оценки качества результирующих изображений

Среднеквадратическая ошибка (σ) — это показатель среднего квадрата отклонений и часто применяется на практике для объективной оценки качества результирующих изображений и вычисляется по формуле

$$\sigma = \frac{1}{NM} \sum_{i=1}^N \sum_{j=1}^M (M_{i,j}^N - M_{i,j}^M)^2,$$

где $N \times M$ — размеры изображения.

Коэффициент сжатия (K_{comp}) — это отношение числа битов, необходимых для представления данных до сжатия, к числу битов, необходимых для представления данных после сжатия и рассчитывается по формуле

$$K_{comp} = \frac{Sum_{original\ image}}{Sum_{compressed\ image}},$$

где $Sum_{original\ image}$ — число битов исходного изображения, $Sum_{compressed\ image}$ — число битов сжатого изображения.

Увеличение коэффициента сжатия приводит к уменьшению битовой скорости передачи при учете ограничений канала и сохранении требуемого качества, и она (битовая скорость) вычисляется по формуле

$$R = \frac{8}{K_{comp}}$$

или по формуле

$$R = \frac{Sum_{compressed\ image}}{NM} \text{ (бит/пиксель)}.$$

Коэффициент сжатия и пиковое отношение сигнал/шум (PSNR) рассчитываются относи-

тельно исходного изображения и сжатого изображения. **Относительное число однородных блоков** (Swe_1) — это доля блоков, не подвергающихся ДКП из-за высокой степени корреляции элементов блока, которая определяется по формуле

$$Swe_1 = \frac{Sum_{nonDCT}}{Sum_{Full}} \%.$$

Данный показатель был учтен при вычислении сложности алгоритмов кодирования изображений и коэффициента сжатия.

Пиковое отношение сигнал/шум (PSNR) — это безразмерная величина, которая аппроксимирует индекс качества изображения, достаточно информативна при сравнительных измерениях, но в плане оценки качества изображений может давать значительные отклонения, выявляемые при субъективной оценке изображения; определяется по формуле

$$PSNR = 10 \lg \frac{255^2}{\sigma}.$$

Важно отметить, при проведении экспериментов обнаружены случаи, когда более низкое значение PSNR может иметь лучшее качество (субъективно) по сравнению с изображением с более высоким отношением сигнал/шум (PSNR).

Сложность алгоритма (W) — это число операций (сложение, умножение, сдвиги и т. д.), приходящихся на один пиксель, определяется по формуле

$$W = \frac{K_{(Mult+Add)} \beta_{Block\ size} k_{Block\ size}}{NM},$$

где $K_{Mult+Add}$ — число сложений и умножений для блока ДКП; $\beta_{Block\ size}$ — число пикселей в блоке и $k_{Block\ size}$ — число блоков $Block\ size$. NM — число пикселей в блоке.

5. Результаты моделирования алгоритмов сжатия изображений

Эксперименты проводились в среде *Microsoft Visual Studio* (C++). Изображения представляют собой снимки с различных цифровых камер наблюдения за дорогами с размерами 256×256 , 512×512 и 1024×1024 .

В работе выбрано три представительных класса транспортных сюжетов из девяти (см. таблицу):

- класс изображений с малой широкополосностью и малой мерой неопределенности ($B_{w1} - U_{n1}$);
- класс изображений со средней широкополосностью и средней мерой неопределенности ($B_{w11} - U_{n2}$);

Таблица сравнения результатов сжатия различными методами

Класс изображений	Критерии оценки	PSNR	K_{comp}	σ	W	R
$(B_{wI} - U_{n1})$	ПКЛ	49,4	35,6	3,4	79	2,4
	JPEG (ДКП)	40,0	27,4	7,5	56	1,7
	ДВП	38,1	36,2	14	64	0,8
	АДКП	42,3	37,3	13	63	0,6
$(B_{wII} - U_{n2})$	ПКЛ	44,3	26,2	2,9	96	2,4
	JPEG (ДКП)	35,2	26,8	17,1	66	3,3
	ДВП	34,5	27,9	19,2	75	1,0
	АДКП	39,9	29,3	15,9	82	0,7
$(B_{wIII} - U_{n3})$	ПКЛ	42,1	21,7	6,8	120	2,9
	JPEG (ДКП)	34,9	22,9	17,7	65	2,2
	ДВП	33,8	24,4	17,6	74	1,1
	АДКП	37,4	28,6	13,6	99	0,6

• класс изображений с большой широкополосностью и большой мерой неопределенности $(B_{wIII} - U_{n3})$.

Остальные сюжеты дали приблизительно такие же результаты. Поскольку было исследовано большое число изображений для каждого класса, то результаты, приведенные в таблице, были усреднены и погрешность при этом не превысила 2 %.

Заключение

Сжатие изображений является чрезвычайно важной частью современных интеллектуальных транспортных систем наблюдения. Имея возможность сжимать изображения до доли их исходного размера, можно сэкономить ценное и дорогое дисковое пространство памяти. Кроме того, передача изображений с мест чрезвычайных ситуаций и аварий в диспетчерский центр управления в реальном времени становится возможной.

Из вышеприведенного исследования можно сделать следующие основные выводы:

1) предлагаемый метод на основе адаптивного косинусного преобразования принимает меньшее значение PSNR и среднее квадратическое отклонение, чем метод Карунена—Лоева, при этом такие показатели, как коэффициент сжатия, сложность и скорость передачи имеют значения меньше других;

2) применение адаптивного способа определения размеров блоков для косинусного преобразования позволило увеличить коэффициент сжатия на 10...20 % при сохранении субъективного качества по сравнению с известными алгоритмами вейвлет и JPEG;

3) поскольку метод Карунена—Лоева является сугубо теоретическим, то для его реализации были вычислены базисные функции

применительно для каждого изображения из представительной выборки транспортных сюжетов по отдельности;

4) получены важные статистические характеристики и зависимости числа блоков, подвергающихся косинусному преобразованию, различного размера от степени однородности областей изображений, позволяющие уменьшить вычислительную сложность устройств передачи видеoinформации в 1,2...3 раза;

5) в качестве перспективного развития направления исследования следует отметить необходимость применения свойства "адаптивности" предложенного алгоритма не только в пространстве сигнала, но и по времени с оценкой скорости движения транспортных средств, а также учета типа движения при формировании различных групп из видеопотока.

Список литературы

1. Rusul A., Hussein D., Sohani L., Saeed A. Applications of Artificial Intelligence in Transport: An Overview. Sustainability 2019, 11, 189; doi: 10.3390/su11010189.
2. Król A. The Application of the Artificial Intelligence Methods for Planning of the Development of the Transportation Network // Transp. Res. Procedia. 2016. Vol. 14. P. 4532—4541.
3. Фахми Ш. С., Цыцулин А. К. и др. Принцип доминантной информации и его приложение к видеoinформатике // Вопросы радиоэлектроники. Сер. Техника телевидения. 2018. Вып. 4. С. 3—16.
4. Hossen M. K., Tuli S. H. A surveillance system based on motion detection and motion estimation using optical flow. In Informatics, Electronics and Vision (ICIEV) // 2016 5th International Conference on IEEE. 2016. P. 646—651.
5. Huang D.-Y., Chen C.-H., Hu W.-C., Yi S.-C., Lin Y.-F. et al. Feature-based vehicle flow analysis and measurement for a real-time traffic surveillance system // Journal of Information Hiding and Multimedia Signal Processing. 2012. Vol. 3. P. 279—294.
6. Porwal S., Chaudhary Y., Joshi J., Jain M. Data Compression Methodologies for Lossless Data and Comparison between Algorithms // International Journal of Engineering Science and Innovative Technology (IJESIT). 2013. Vol. 2, N. 2. P. 142—147.
7. Vijayvargiya S. G., Pandey R. P. A Survey: Various Techniques of Image Compression // International Journal of Computer Science and Information Security. 2013. Vol. 11, N. 10.
8. Фахми Ш. С. Концепция проектирования интеллектуальных транспортных видеосистем на основе технологии "система на кристалле" // Журнал университета водных коммуникаций. 2013. Вып. II (XVIII). С. 79—88.
9. Ричардсон Ян. Видеокодирование. H.264 и MPEG-4 — стандарты нового поколения. М.: Техносфера, 2005. 368 с.
10. Фахми Ш. С., Цыцулин А. К., Колесников Е. И., Очкур С. В. Функционал взаимнообмена сложности и точности систем кодирования непрерывного сигнала // Информационные технологии. 2011. № 4. С. 71—77.
11. Моисеев Н. Н. Математические задачи системного анализа. М., Наука, 1981. 488 с.
12. Castrillon-Candas J. E., Amaratunga K. Fast Estimation of Continuous Karhunen-Loeve Eigenfunctions Using Wavelets // IEEE Transactions on Signal Processing. 2002.
13. Mulla A., Baviskar J., Baviskar A., Warty C. Image compression scheme based on zig-zag 3D-DCT and LDPC coding // Proc. International Conference on Advances in Computing, Communications and Informatics. 2014. P. 2380—2384.
14. Thirumoorthi C., Karthikeyan T. A study on discrete wavelet transform compression algorithm for medical images // Biomedical Research. 2017. Vol. 28 (4). P. 1574—1580.

15. **Фахми Ш. С., Альмахрук М. М.** и др. Точность, скорость и сложность устройств кодирования изображений по опорным точкам // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16, № 4. С. 678–689.

16. **Abdelhalim M. B., Salama A. E.** Implementation of 3D DCT based Video Encoder Decoder System" // IEEE International Symposium on signals, Circuits and Systems. 2003. Vol. 2. P. 389–392.

17. **Saponara S., Fanucci L., Terreni P.** Low- Power VLSI architectures for 3D Discrete Cosine Transform" // IEEE 48th Midwest Symposium on Circuits and Systems. 2013. Vol. 3. P. 1567–1570.

18. **Fan Y., Shan-Ann C., Kuo-Gi W., Jun-Lin Y.** 3D-DCT Chip Design for 3D Multi-view Video Compression // Appl. Math. Inf. Sci. 2012. Vol. 6, N. 2S. P. 567S–572S.

19. **Фахми Ш. С., Зубакин И. А.** Классификация нестационарных изображений и разработка методики оценки алгоритмов кодирования источника // Науч.-техн. вестник СПбГУ ИТМО. 2010. № 2 (66). С. 54–59.

20. **Btoush M. H., Dawahdeh Z. E.** A Complexity Analysis and Entropy for Different Data Compression Algorithms on Text Files // Journal of Computer and Communications. 2018. Jan. 12. P. 301–315. DOI: 10.4236/jcc.2018.61029

S. Sh. Fahmi, Professor, e-mail: shakeebf@mail.ru, St. Petersburg state electrotechnical University V. I. Ulyanov (Lenin) "LETI", Solomenko Institute of Transport Problems of the Russian academy of sciences, **N. G. Ryjov**, Assistant Professor, e-mail: ngryzhov@etu.ru, St. Petersburg state electrotechnical University V. I. Ulyanov (Lenin) "LETI", **Y. A. Hasan**, post-graduate student, e-mail: midocom@mail.ru, St. Petersburg state electrotechnical University. V. I. Ulyanov (Lenin) "LETI", **E. S. Kalinina**, Assistant Professor, e-mail: elenakali21@yandex.ru, Saint-Petersburg University of state fire service of Emercom of Russia

Adaptive Algorithm of Image Encoding and Decoding in Transport Video Surveillance Systems

It is well known that the image processing standards (JPEG) and video processing standards (MPEG), based on the use of spectral cosine transformation, do not meet the growing requirements for high quality video transmission obtained from surveillance cameras in transport. The reason for this is the large amount of visual data and the fact that the optimization of the video codec should be done (at least) on three parameters: bit rate, distortion and computational complexity. Therefore, the task arises to develop new image compression algorithms that are best aligned with existing standards. One of the algorithms for improving the process of coding and decoding is considered in this paper. It differs from the existing ones in that the sizes of blocks subjected to cosine transformation can be arbitrary and are selected taking into account the specifics of transport subjects.

Keywords: Image encoding, cosine transformation, adaptability, transport, transfer rate, complexity

DOI: 10.17587/it.25.474-481

References

1. **Rusul A., Hussein D., Sohani L., Saeed A.** Applications of Artificial Intelligence in Transport: An Overview. Sustainability 2019, 11, 189; DOI:10.3390/su11010189.

2. **Król A.** The Application of the Artificial Intelligence Methods for Planning of the Development of the Transportation Network, *Transp. Res. Procedia*, 2016, vol. 14, pp. 4532–4541.

3. **Fahmi S. Sh., Sycolin A. K.** et al. the dominant information Principle and its application to videoinformation, *Questions of radio electronics, ser. The technique of television*, 2018, iss. 4, pp. 3–16 (in Russian).

4. **Hossen M. K., Tuli S. H.** A surveillance system based on motion detection and motion estimation using optical flow. In Informatics, Electronics and Vision (ICIEV), 2016 5th International Conference on IEEE, 2016, pp. 646–651.

5. **Huang D.-Y., Chen C.-H., Hu W.-C., Yi S.-C., Lin Y.-F.** et al. Feature-based vehicle flow analysis and measurement for a real-time traffic surveillance system, *Journal of Information Hiding and Multimedia Signal Processing*, 2012, vol. 3, pp. 279–294.

6. **Porwal S., Chaudhary Y., Joshi J., Jain M.** *International Journal of Engineering Science and Innovative Technology (IJESIT)*, 2013, vol. 2, no. 2, pp. 142–147.

7. **Vijayvargiya S. G., Pandey R. P.** *International Journal of Computer Science and Information Security*, 2013, vol. 11, no. 10.

8. **Fahmi S. Sh.** *Journal of University of water communications*, 2013, iss. II (XVIII), pp. 79–88 (in Russian).

9. **Richardson I.** Video coding. H. 264 and MPEG — 4-new generation standards, Moscow, Technosphere, 2005, 368 p. (in Russian).

10. **Fahmi S. Sh., Sycolin A. K., Kolesnikov E. I., Ochkur S. V.** *Information Technology*, 2011, no. 4, pp. 71–77 (in Russian).

11. **Moiseev N. N.** Mathematical problems of system analysis, Moscow, Science, 1981, 488 p. (in Russian).

12. **Castrillon-Candas J. E., Amaratunga K.** Fast Estimation of Continuous Karhunen-Loeve Eigenfunctions Using Wavelets, *IEEE Transactions on Signal Processing*, 2002.

13. **Mulla A., Baviskar J., Baviskar A., Warty C.** Image compression scheme based on zig-zag 3D-DCT and LDPC coding, *Proc. International Conference on Advances in Computing, Communications and Informatics*, 2014, pp. 2380–2384.

14. **Thirumoorthi C., Karthikeyan T.** A study on discrete wavelet transform compression algorithm for medical images, *Bio-medical Research*, 2017, vol. 28 (4), pp. 1574–1580.

15. **Fahmi S. Sh.** *Journal of University of Water Communications*, 2013, iss. II (XVIII), pp. 79–88 (in Russian).

16. **Abdelhalim M. B., Salama A. E.** Implementation of 3D DCT based Video Encoder Decoder System", *IEEE International Symposium on signals, Circuits and Systems*, 2003, vol. 2, pp. 389–392.

17. **Saponara S., Fanucci L., Terreni P.** Low- Power VLSI architectures for 3D Discrete Cosine Transform", *IEEE 48th Midwest Symposium on Circuits and Systems*, 2013, vol. 3, pp. 1567–1570.

18. **Fan Y., Shan-Ann C., Kuo-Gi W., Jun-Lin Y.** 3D-DCT Chip Design for 3D Multi-view Video Compression, *Appl. Math. Inf. Sci.*, 2012, vol. 6, no. 2S, pp. 567S–572S.

19. **Fahmi S. Sh., Zubakin I. A.** Classification of non-stationary images, and developing methods of evaluation algorithms of source coding, *Nauch. technical Bulletin of SPBU ITMO*, 2010, no. 2 (66), pp. 54–59 (in Russian).

20. **Btoush M. H., Dawahdeh Z. E.** A Complexity Analysis and Entropy for Different Data Compression Algorithms on Text Files, *Journal of Computer and Communications*, 2018, Jan. 12, pp. 301–315, DOI: 10.4236/jcc.2018.61029

В. Е. Гвоздев, д-р техн. наук, проф., wega55@mail.ru,
О. И. Христовуло, д-р техн. наук, проф., o-hristodulo@mail.ru,
Уфимский государственный авиационный технический университет

Информационная поддержка принятия решений в задачах управления отходами на основе математико-геоинформационного моделирования*

Излагаются основные составляющие проблемы управления твердыми коммунальными отходами с позиции рассмотрения системы управления отходами как открытой сложной системы с нечеткими целями. Выделены системообразующие факторы формирования территориально-распределенной системы управления отходами. Рассматриваются функциональные задачи системы информационной поддержки управления отходами, приведена структура информационной системы. Предложен подход к решению задачи районирования территориальной системы на основе как частных, так и по совокупности характеристик состояния, с использованием математико-геоинформационного моделирования. Приведен пример решения задачи районирования территории Республики Башкортостан по частным и совокупности показателей, характеризующих воздействие отходов на состояние природной среды и здоровье населения.

Ключевые слова: система управления отходами, информационная поддержка принятия решений, математико-геоинформационное моделирование, районирование территории по частным и совокупным показателям

Введение

Современное развитие экономики Российской Федерации характеризуется ростом объема и уровня промышленного производства. Вместе с увеличением произведенной промышленностью продукции естественным образом растет количество отходов производства, большая часть которых отправляется к местам захоронения или длительного хранения.

Геотехнические системы, представляющие собой неделимое единство природного, техногенного и социального компонентов, являются разновидностью сложных систем. Это обстоятельство служит основанием для научно обоснованной адаптации признанных методов и подходов, используемых при управлении сложными системами, применительно к реализации стратегий, программ и проектов, связанных с управлением отходами на территории промышленно-развитых регионов РФ.

Одна из задач многоаспектной проблемы управления отходами — оценивание текущего

экологического состояния исследуемой территории с точки зрения населения.

В статье описаны особенности систем управления отходами; выделены системообразующие факторы, позволяющие сформировать территориально-распределенную систему управления отходами; определены локальные контуры управления; сформулированы цель и задачи координации управления в локальных контурах применительно к системе управления отходами; представлена технологическая основа информационной поддержки управления отходами, включая математико-геоинформационный анализ показателей состояния системы управления отходами. Приведены примеры решения одного из классов задач математико-геоинформационного анализа — районирование территории Республики Башкортостан по частным показателям, характеризующим влияние отходов на компоненты окружающей среды.

Особенности систем управления отходами

Система управления твердыми коммунальными отходами (далее ТКО) относится к классу открытых сложных систем с нечеткими целями

* Работа выполнена при поддержке РФФИ, грант № 18-08-00885-а.

ми. В управление ТКО вовлечены различные заинтересованные стороны, имеющие различные мотивы и разные персональные онтологические модели проблемной ситуации, обусловленной возникновением, хранением, транспортировкой и переработкой отходов. Это делает необходимым совершенствование методов выработки консолидированных решений, признаваемых всеми заинтересованными сторонами, что создает основу скоординированной деятельности различных государственных органов, уполномоченных на решение комплекса задач, связанных с проблемой управления ТКО. Методологическую основу получения консолидированных решений в условиях нечетких целей управления и мотивов неоднородных акторов, вовлеченных в решение управленческих задач, могут составить положения теории конвергентного управления и эвергетики [2, 3].

Управление отходами приходится осуществлять не только в условиях нечетких целей управления и различия мотивов представителей государственных структур, бизнеса и населения в урегулировании проблемной ситуации, но и при наличии фундаментальной и измерительной неопределенности, обусловленной недостатком знаний о механизмах влияния отходов на состояние территориальных систем; недостаточной изученностью причинно-следственных связей между состоянием отходов и состоянием окружающей природной среды и здоровьем населения; ограниченностью данных и информации, присутствующих в ведомственных информационных системах и так или иначе связанных с возникновением, сбором, хранением, переработкой отходов. Отмеченные обстоятельства обуславливают необходимость выработки консолидированного решения по урегулированию проблемной ситуации с привлечением аксиологических знаний акторов, вовлеченных в урегулирование проблемной ситуации [2]. Элементами коммуникационной платформы при этом являются структурированные персональные знания субъектов, а также результаты разноаспектного анализа данных и информации, включая геоданные и геоинформацию [3], существующих в ведомственных информационных системах, уполномоченных на решение разноплановых задач, связанных с обеспечением экологической, техногенной безопасности и управления состоянием здоровья населения.

К системообразующим факторам, позволяющим выделить территориально-распределенную систему управления отходами, следует отнести:

а) объемы и особенности промышленных и бытовых отходов, возникающих, хранящихся

и перерабатываемых на различных участках территориальной системы;

б) территориально-временную динамику воздействия отходов на состояние природной среды и здоровье населения;

в) территориально-временные аспекты сбора, хранения, транспортировки и переработки отходов;

г) территориально-временные особенности отношения населения к проблеме управления отходами.

В качестве локальных контуров управления в системе управления отходами следует выделить:

а) управление негативным воздействием отходов на состояние окружающей среды и здоровье населения;

б) управление выбором мест постоянного и временного хранения отходов, мест строительства предприятий по переработке отходов с учетом соотношения объемов производимых и уже находящихся в местах хранения отходов разных видов (промышленных, бытовых), а также состояние транспортной сети на различных участках территориальной системы;

в) управление требованиями к деятельности предприятий в части переработки отходов;

г) управление экономическими последствиями хранения и переработки отходов, включая назначение тарифов на сбор, хранение, транспортировку и переработку отходов;

д) управление ценностями, стремлениями, поведением населения в вопросах управления отходами при одновременном изменении процессов, методов, стратегий, структуры системы управления отходами.

Приведенный перечень не является окончательным, его состав может корректироваться по мере накопления фундаментальных и аксиологических знаний, получаемых в ходе практической деятельности.

Для каждого локального контура управления определяются целевые группы клиентов и ассоциированные с ними цели управления; показатели результативности и эффективности управления; показатели качества управления. Подходом к координации целей управления в локальных контурах является взаимное согласование желаний и ожиданий неоднородных акторов. Подходами к координации процессов, соотносимых с локальными контурами управления, являются сбалансированное сочетание:

- прямой координации (начальник—подчиненный);
- стандартизации целей (установления целей для исполнителей, которые самостоятельно

выбирают средства их достижения с тем, чтобы достигнуть заданных целевых показателей управления);

- стандартизации деятельности в рамках реализации процессного подхода;
- стандартизации ценностей населения для повышения культуры населения в вопросах обращения с бытовыми отходами.

Построение конвергентного когнитива, содержанием которого является консолидированное мнение целевых групп акторов (население, государственные служащие, бизнесмены, общественные деятели, научное сообщество), вовлеченных в решение проблемы управления отходами и имеющих вместе с тем разную мотивацию, предполагает решение следующих классов задач:

- определение содержания проблемной ситуации, выделение локальных целей управления в каждом из контуров управления;
- построение на основе онтологических моделей акторов онтологических моделей групп акторов и построение на их основе посредством технологий конвергентного управления признаваемую всеми онтологической модели проблемной ситуации;
- разработка на основе онтологической модели ситуации стратегии управления отходами на территории, попадающей в зону ответственности системы управления; разработка на ее основе в рамках каждого локального контура управления стратегий управления отходами, дорожных карт реализации локальных стратегий, механизмов реализации составляющих дорожных карт.

Построение конвергентного когнитива начинается с выбора и идентификации доступной для формализации части проблемной ситуации, основу чего составляет разноаспектный анализ данных, содержащихся в различных источниках (официальных документах, отчетах, научных статьях и т.д.), а также изучения знаний экспертов.

Критически важными задачами управления являются:

а) выделение самоусиливающихся процессов роста, создающих основу глубинных перемен в системе управления отходами;

б) выявление процессов, тормозящих позитивные изменения, определение причин и ограничивающих факторов, противодействующих изменениям; выработка подходов и реализация действий по парированию / устранению ограничивающих факторов.

Основной целью координации управления в локальных контурах является обеспечение

консонанса самоусиливающихся процессов в локальных контурах управления.

Сложность решения слабоформализованных задач, связанных с построением локальных систем управления, обусловлена:

- необходимостью обоснования отбора в исходных описаниях территориальных систем необходимых знаний, позволяющих выделить системы управления из окружающей среды;
- необходимостью систематизации знаний о составных частях систем управления отходами и отражения их в смысловых понятиях, в базе которых будут в последующем строиться онтологические модели, которые составят коммуникационную основу решения задач управления отходами;
- необходимостью структуризации знаний о составных частях системы управления отходами для последующего представления их в виде структурированных формализованных моделей (когнитивных моделей).

Ситуационное управление — перспективный подход к решению задач управления отходами в реальном масштабе времени. Реализация ситуационного управления сводится к двум основным процедурам:

1) выявление на основе анализа онтологической модели ситуации [2] архетипа проблемной ситуации [3, 9];

2) соотнесение возникшей проблемной ситуации с содержимым ранее сформированной базы данных о способах урегулирования аналогичных проблемных ситуаций и полученных при этом результатах.

Ситуационный подход предполагает использование заранее подготовленных сценариев "ситуация—решение", которые затем выбираются субъектами управления в соответствии с текущими ситуациями. При решении задач управления распределенными системами, к числу которых относятся задачи управления отходами, реализация ситуационного подхода осложняется тем, что в силу территориальных особенностей воздействия отходов на состояние природной среды и здоровье населения решения, эффективные на одних участках территории, могут привести к негативным результатам на других участках. Кроме этого, в силу территориальных особенностей субъективной составляющей в решении задач управления отходами является, например, отношение населения городских и сельских муниципальных образований к сбору, транспортировке и переработке отходов.

Совокупность выделенных факторов во многих случаях приводит к отказу от моделирования как инструмента информацион-

ной поддержки управления, и решение задач управления полностью остается за субъектом-оператором. Из-за отсутствия капитального информационного фундамента способы решения задач управления субъектами-операторами являются интуитивными и во многих случаях в итоге оказываются нерациональными.

Технологическая основа информационной поддержки управления отходами

Основой информационной поддержки выработки консолидированных решений по регулированию проблемной ситуации является представление в структурированном виде данных, информации и знаний, характеризующих негативное влияние отходов на состояние окружающей среды и здоровье населения. Реализация этой деятельности создает основу формирования альтернатив решения задач, связанных со сбором, организацией временного размещения, транспортированием, захоронением, утилизацией и переработкой отходов. Технологическую основу создания информационной платформы системы управления составляют технологии визуализации и разноаспектного анализа геоданных и геоинформации; технологии выделения причинно-следственных связей и анализа структур, соответствующих различным точкам зрения на проблему управления отходами; технологии построения когнитивных карт и реализации когнитивного моделирования; технологии конвергентного управления; технологии математико-статистического анализа данных, в том числе малых по объему и низких по точности; технологии обработки нечетких и размытых данных. В данной работе фокусом обсуждения являются технологии математико-геоинформационного моделирования, основу которых применительно к задачам управления отходами составляет системное сочетание геоинформационных технологий и методов математико-статистического моделирования. Использование технологий геоинформационного моделирования создает основу для представления в сопоставимом виде (в виде тематических картографических слоев) информации и данных, характеризующих состояние природных компонентов геотехнических объектов; характеристик мест возникновения отходов различного происхождения (производственных, бытовых) на разных участках территориальной системы; мест размещения и захоронения отходов; характеристик транспортной сети. Использование математико-статистических методов создает фундамент ана-

лиза территориально-временной изменчивости показателей состояния компонентов территориальных систем в условиях неопределенности свойств взаимосвязанных процессов, обусловленных природными, социальными, технологическими факторами, в том числе процессами возникновения, хранения, транспортировки и переработки ТКО.

Информационная поддержка управления отходами должна обеспечивать единую информационно-технологическую цепочку сбора, хранения, обработки информации о размещении промышленных отходов и включать информационно-аналитическую базу данных, развивающиеся банки данных и систему обработки электронных векторных карт территории.

Функциональные возможности системы информационной поддержки принятия решений по управлению отходами должны обеспечивать для всех неоднородных акторов, причастных к проблеме управления отходами на территории субъекта РФ:

1) пополнение и редактирование справочника природопользователей, в том числе промышленных предприятий, участвующих в обращении с отходами;

2) пополнение классификационного каталога отходов видами отходов, которые образуются, транспортируются, утилизируются и хранятся на территории субъекта РФ;

3) анализ и вывод отчетов об объектах экологического мониторинга и потенциальной экологической опасности;

4) поддержание карт об объектах размещения промышленных отходов, предприятиях по переработке и утилизации отходов в актуальном состоянии и оперативное формирование необходимого картографического материала;

5) анализ и обработка пространственной информации для формирования карт пригодности к размещению объектов промышленных отходов;

6) анализ территориальных аспектов, нагрузки, оказываемой источниками отходов и местами хранения отходов на природную среду и состояние здоровья населения.

На рис. 1 приведена структура информационной системы управления отходами, которая реализована в Министерстве природопользования и экологии Республики Башкортостан. Одной из функциональных задач, решаемых представленной системой, является задача районирования территории Республики Башкортостан по частным и совокупным показателям, характеризующим воздействие отходов на состояние природной среды и здоровье населения.

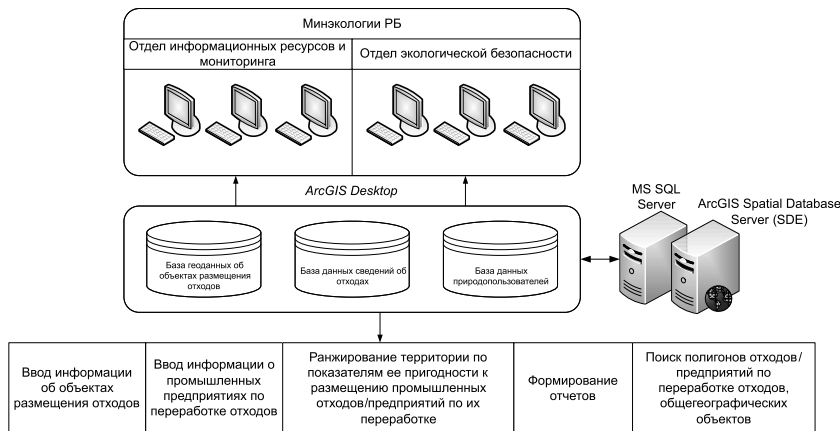


Рис. 1. Структура информационной системы управления отходами

Наименование	Содержание показателя	Локальный контур управления
Количество отходов на душу населения/км ² территории	Негативная нагрузка на состояние природной среды и здоровье населения	Управление негативным воздействием на состояние природной среды и здоровье населения
Количество использованных и обезвреженных отходов на душу населения/км ² территории	Парирование негативных воздействий отходов на состояние окружающей среды и здоровье населения	<ul style="list-style-type: none"> • Управление негативным воздействием на состояние природной среды и здоровье населения • Управление требованиями к деятельности предприятий в части переработки отходов

Методическая основа решения задачи районирования по показателям состояния территориальной системы

Основу оценки состояния системы управления отходами составляет территориально-временной анализ симптомов, косвенно характеризующих процессы, управление которыми осуществляется в локальных системах управления [12–14].

В таблице в качестве примера приводится описание отдельных показателей, косвенно характеризующих состояние территориальной системы с точки зрения негативного влияния отходов.

В работах, посвященных исследованию сложных систем, отмечается, что решение задачи классификации составляет основу анализа гетерогенных динамических объектов [1, 4, 6–8, 11]. Применительно к анализу состояния территориальных систем это выражается в классификации участков территории (в задаче управления отходами — муниципальных образований) по значениям характеристик состояния [10, 12, 15, 16]. Классификация участков территории основана на разработке формальных методов регуляризации пространствен-

ных данных, позволяющих отнести каждый участок территории к тому или иному классу состояний.

В общем случае задача классификации представляется в виде

$$\varphi(x, y) \rightarrow S(x, y), \quad (1)$$

где $\varphi(x, y)$ — рельеф поверхности, образуемой характеристикой состояния; $S(x, y)$ — пространственная мозаика, образуемая участками исследуемой территории, причем каждому элементу мозаики ставится в соответствие определенный класс состояния территориальной системы [5].

Основой преобразования $\varphi(x, y)$ в $S(x, y)$ является использование различных шкал, устанавливающих соответствие между диапазонами значений показателей рельефа характеристик и классами состояния территории, т. е.

$$\alpha_i \leq \varphi(x, y) \leq \beta_i \rightarrow C_i, \quad (2)$$

где α_i, β_i — границы i -го подынтервала классификационной шкалы; C_i — i -й класс состояния участка территории.

При решении практических задач, связанных с оценкой состояния территориальной системы, достаточно типичной является ситуация, когда вместо $\varphi(x, y)$ известны лишь выборочные значения рельефа характеристик состояния $Z(x_i, y_i) = \varphi(x_i, y_i), (i = \overline{1; N})$, соответствующие узлам неравномерной сетки измерений. В этом случае модель (1) преобразуется к виду

$$Z(x_i, y_i) \xrightarrow{P_\alpha} \hat{S}(x, y), (i = \overline{1; N}), \quad (3)$$

где $\hat{S}(x, y)$ — оценка $S(x, y)$, зависящая от N , местоположения узлов сетки измерений и выбранного способа P_α преобразования рельефа в пространственную мозаику.

При решении прикладных задач классификации территориальных систем достаточно часто приходится сталкиваться с ситуацией, когда классификационная шкала априорно неизвестна. В этом случае \hat{P}_α должна стремиться по фактически доступным выборочным данным.

В работах [2–5] отмечается, что если $Z(x_i, y_i)$ представляют собой случайные величины, интегральной характеристикой статистического рельефа является функция распределения

$\hat{F}(z)$. В связи с этим процедура классификации должна строиться с учетом особенностей оценки $\hat{F}(z)$, получаемой по значениям $Z(x_i, y_i) (i = \overline{1; N})$, и модель (3) преобразуется к виду

$$\begin{array}{c} \hat{F}(z) \\ \nearrow \quad \searrow \\ Z(x, y) \rightarrow \hat{S}(x, y) \end{array} \quad (4)$$

Из этой модели следует, что качество мозаики $\hat{S}(x, y)$ определяется близостью оценки $\hat{F}(z)$ к закону распределения $F(z)$, которому соответствует выборка $Z(x_i, y_i)$.

Обобщенная схема классификации участков территориальной системы по выборочным данным основана на следующем:

- вся информация о состоянии территориальной системы содержится в выборочных данных, причем оценка $\hat{F}(z)$ является исчерпывающей (в смысле параметра z) характеристикой состояния;
- оценивание пространственной мозаики $\hat{S}(x, y)$ сводится к разделению территории на участки, причем каждый из участков соотносится с одним из классов состояния. Участкам, принадлежащим к одному и тому же классу, ставится в соответствие одинаковое значение характеристики рельефа, по которым может быть построена интегральная оценка $F_{\hat{S}}(z)$.

Таким образом, число классов состояния и границы непересекающихся классов должны выбираться таким образом, чтобы соблюдалось условие:

$$P_j [\hat{F}(z), F_{\hat{S}}(z)] \rightarrow \min, \quad (5)$$

где P_j — j -е правило сравнения оценок $\hat{F}(z)$, $F_{\hat{S}}(z)$.

Если исходить из того, что выборочные значения $Z(x_i, y_i), i = \overline{1; N}$, являются случайными величинами, которым соответствует функция распределения $\hat{F}(z)$, для формализации процедуры построения классификационной шкалы целесообразно использовать подход, основанный на группировании статистических данных. В работе [11] описываются различные подходы к построению оценок $\hat{F}(z)$ по выборкам конечного (в том числе малого) объема, а также разные подходы к решению задачи (5). Там же приводится описание инженерной методики, позволяющей полностью формализовать процедуру классификации территориальной системы на основе однородных выборок $Z(x_i, y_i), i = \overline{1; N}$.

Подходы к решению задачи классификации участков территориальной системы по показателям состояния должны увязываться с уровнями управления: тематическим, операционным. В связи с этим целесообразно ввести понятия "абсолютная" "относительная" классификационные шкалы. Целью построения "относительной" классификационной шкалы является представление в наглядной картографической форме текущей (оперативной) обстановки, соответствующей временному срезу. При этом в качестве элементов выборки выступают данные $Z^{(l)}(x_i, y_i), i = \overline{1; N}$, соответствующие l -м условиям исследований.

Целью построения "абсолютной" классификационной шкалы является создание основы для сопоставления состояния территориальной системы в разных временных срезах.

Абсолютная шкала формируется на основе данных, соответствующих совокупности временных срезов, и предназначена для сопоставления состояний территорий в различных временных срезах ($l = 1, 2, \dots, h$). В качестве элементов выборки в этом случае выступает объединенная выборка

$$\{z\} = \bigcup_{l=1}^h z^{(l)}(x_i, y_i), i = \overline{1; N}. \quad (6)$$

Ограничениями описанного подхода является необходимость соблюдения условия однородности выборочных данных, регистрируемых в узлах сетки измерений.

Рассмотрим пример решения задачи зонирования территории Республики Башкортостан в рамках разработанной информационной системы управления отходами по реальным значениям следующих показателей, косвенно характеризующих состояние территориальной системы с точки зрения негативного влияния отходов:

- 1) количество образованных отходов на душу населения, т;
- 2) количество образованных отходов на 1 км², т;
- 3) количество использованных и обезвреженных отходов на душу населения, т;
- 4) количество использованных, обезвреженных отходов на 1 км², т;

На рис. 2 (см. третью сторону обложки) приведены результаты районирования территории республики по нескольким частным показателям состояния территории Республики Башкортостан за 2015 год. В целях обеспечения сопоставимости результатов, соответствующих различным показателям, муниципальные районы были разделены на три класса, которые

соответствовали следующим признакам, отражающим состояние территории Республики Башкортостан с точки зрения негативного влияния отходов: "низкий уровень негативного воздействия"; "средний уровень негативного воздействия" и "высокий уровень негативного воздействия".

Результаты районирования территориальных систем по частным показателям состояния создают основу для оценки эффективности системы управления отходами.

В литературе приведено соотношение для расчета парного коэффициента корреляции между i -м и j -м признаками [5]:

$$r^{(i,j)} = \sin\left(\frac{\pi}{2} q^{(i,j)}\right).$$

Здесь $q^{(i,j)}$ определяется по правилу

$$q^{(i,j)} = p^{(i,j)} - q^{(i,j)},$$

где $p^{(i,j)}$ — доля совпадающих значений признаков, $q^{(i,j)}$ — доля несовпадающих значений признаков.

Применительно к результатам, представленным на рис. 2, $p^{(i,j)}$ — доля муниципальных образований, у которых классы состояний по разным признакам совпадают. Соответственно, $q^{(i,j)}$ — доля муниципальных образований, у которых значения признаков не совпадают.

Значение парного коэффициента корреляции для признаков "количество образованных отходов на душу населения / количество использованных и обезвреженных отходов на душу населения" составило $r^{(i,j)} = -0,5$. Для признаков "количество образованных отходов на 1 км² территории / количество использованных и обезвреженных отходов на 1 км² территории" значение парного коэффициента корреляции составило $r^{(i,j)} = -0,12$.

Результаты расчетов позволяют сделать заключение о недостаточной эффективности системы управления отходами. Это может служить исходной точкой (симптомом) для последующего углубленного анализа причин недостаточной эффективности системы управления, например, посредством анализа коренных причин возможных нежелательных событий — RCA (RootCauseAnalysis).

Заключение

Предлагаемый подход к информационной поддержке принятия решений в задачах управления отходами на основе математико-

геоинформационного моделирования позволяет сделать следующие выводы:

1. Необходимым условием повышения обоснованности управленческих решений в условиях фундаментальной неопределенности, недостатка объективных знаний о процессах, протекающих в сложных системах, противоречивости целей, соответствующих локальным контурам управления, является обеспечение доступности результатов разноаспектного анализа данных и сведений, поступающих из различных источников, для всех неоднородных акторов, причастных к решению задач управления отходами.

2. Картографические модели создают основу для обеспечения сопоставимости и комплексного анализа разнородных данных и информации, поступающей из различных источников. Картографическая визуализация первичных данных, информации и аксиологических знаний субъектов, причастных к решению различных задач управления отходами, создает основу для принятия консолидированных решений в условиях неопределенности состояния объекта управления и нечеткости целей управления.

3. Результаты математико-геоинформационного моделирования составляют основу для разноаспектного анализа эффективности управления состоянием территориальных систем.

Результаты исследований, представленные в статье, частично поддержаны грантом РФФИ 18-08-00885-а "Методологические основы многокритериального управления процессом выбора местоположения промышленных предприятий по переработке отходов на основе положений эвергетики".

Список литературы

1. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика. Классификация и снижение размерности. М.: Финансы и статистика, 1989. 607 с.
2. Виттих В. А. Управление ситуациями в сложных развивающихся системах с применением интересубъективных теорий // Мехатроника, автоматизация, управление. 2011. № 12. С. 2—6.
3. Райков А. Н. Конвергентное управление и поддержка решений. М.: ИКАР, 2009. 244 с.
4. Бочаров М. К. Методы математической статистики в географии. М.: Мысль, 1971. 371 с.
5. Мирский Г. Я. Характеристики стохастической взаимосвязи и их изменения. М.: Энергоиздат, 1982. 320 с.
6. Гиг Дж. Ван. Прикладная общая теория систем. М.: Мир, 1981. Кн. 1. 341 с., Кн. 2. 730 с.
7. Губанов В. А., Захаров В. В., Коваленков А. Н. Введение в системный анализ. Л.: Изд-во ЛГУ, 1988. 232 с.
8. Дэйвисон М. Многомерное шкалирование: Методы наглядного представления данных / Пер. с англ. В. С. Каменского. М.: Финансы и статистика, 1988. 254 с.
9. Сенге П. М. Пятая дисциплина: искусство и практика самообучающейся организации. М.: ЗАО "Олимп Бизнес", 1999.

10. Михайлов Н. И. Физико-географическое районирование. М.: МГУ, 1985. 184 с.

11. Гузаиров М. Б., Гвоздев В. Е., Ильясов Б. Г., Колоденкова А. Е. Статистическое исследование территориальных систем. М.: Машиностроение, 2008. 187 с.

12. Христовуло О. И., Салимзянов И. Ф., Гареева Н. Р. Разработка информационной системы размещения объектов техногенной опасности с использованием нечеткой логики // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. 2015. Т. 5 (229). С. 47–58.

13. Khristodulo O., Davlethakova Z., Gvozdev V. Spatial Information Processing for Decision-making Support of Siting Sources of Technogenic Hazards Using Computer Technologies // 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). Челябинск, 2017. P. 1–5.

14. Гвоздев В. Е., Христовуло О. И., Фахретдинова Э. Б. Информационная поддержка управления отходами на основе когнитивного, геоинформационного и математического моделирования // Сб. науч. Тр. Междунар. науч.-техн. Конф. Перспективные информационные технологии. Самара, 2018. С. 1110–1114.

15. Бахонина Е. И., Савоськина Р. Р. Ситуация с твердыми бытовыми отходами в Республике Башкортостан // Изв. Самар. Науч. Центра РАН. 2014. № 1 (6). С. 1689–1694.

16. Об осуществлении регионального государственного экологического надзора на территории Республики Башкортостан в 2014 году: гос. докл. М-во природопользования и экологии Республики Башкортостан. Уфа, 2015. 150 с.

V. E. Gvozdev, D. Sc., Professor, wega55@mail.ru,

O. I. Khristodulo, D. Sc., Professor, o-hristodulo@mail.ru,

Ufa State Aviation Technical University, Ufa, 45008, Russian Federation

Information Support of Decision-Making in Waste Management Problems on the Basis of Mathematical and Geoinformation Modeling

The main components of the problem of municipal solid waste management are described from the point of view of the waste management system as an open complex system with fuzzy goals. The system-forming factors of the formation of a geographically-distributed waste management system are identified. The functional tasks of the waste management information support system are considered, the structure of the information system is given. An approach is proposed to solve the problem of zoning of the territorial system on the basis of both private and aggregate characteristics of the situation, based on mathematical geoinformation modeling. An example is given of solving the problem of zoning the territory of the Republic of Bashkortostan according to the private and aggregate of indicators characterizing the impact of waste on the environment and public health.

Keywords: waste management system, information support of decision-making, mathematical and geoinformation modeling, zoning of the territory by private and aggregate indicators

DOI: 10.17587/it.25.482-489

References

1. Aivazyan S. A., Buchstaber V. M., Enyukov I. S., Meshalkin L. D. *Applied statistics. Classification and dimension reduction*, Moscow, Finansy i statistika, 1989, 607 p. (in Russian).

2. Vittikh V. A. *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2011, no. 12, pp. 2–6 (in Russian).

3. Raikov A. N. *Convergent management and solution support*, Moscow, IKAR, 2009, 244 p. (in Russian).

4. Bocharov M. K. *Methods of mathematical statistics in geography*, Moscow, Mysl', 1971, 371 p. (in Russian).

5. Mirsky G. Ya. *Characteristics of stochastic interrelation and their changes*, Moscow, Energoizdat, 1982, 320 p. (in Russian).

6. Gig Dzh. Van. *Applied General theory of systems*, Moscow, Mir, 1981, vol. 1, 341 p., vol. 2, 730 p. (in Russian).

7. Gubanov V. A., Zaharov V. V., Kovalenkov A. N. *Introduction to system analysis*, Leningrad, Publishing house of LGU, 1988, 232 p. (in Russian).

8. Dejvison M. *Multidimensional scaling: Methods of visual representation of data*, Moscow, Finansy i statistika, 1988, 254 p. (in Russian).

9. Senge P. M. *The fifth discipline: the art and practice of self-learning organization*, Moscow, ZAO "Olimp Biznes", 1999 (in Russian).

10. Mihajlov N. I. *Physical and geographical zoning*, Moscow, MGU, 1985, 184 p. (in Russian).

11. Guzaиров M. B., Gvozdev V. E., Il'yasov B. G., Kolodenkova A. E. *A statistical study of territorial systems: monograph*, Moscow, Mashinostroenie, 2008, 187 p. (in Russian).

12. Khristodulo O. I., Salimzyanov I. F., Gareeva N. R. *Nauchno-Tekhnicheskie Vedomosti SPbGPU. Informatika. Telekomunikacii. Upravlenie*, 2015, vol. 5 (229), pp. 47–58 (in Russian).

13. Khristodulo O., Davlethakova Z., Gvozdev V. *2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2017, pp. 1–5.

14. Gvozdev V. E., Khristodulo O. I., Fahretdinova E. B. *Sbornik nauchnyh trudov Mezhdunarodnoj nauchno-tekhneskoj konferencii Perspektivnye informacionnye tekhnologii*, 2018, pp. 1110–1114 (in Russian).

15. Bahonina E. I., Savos'kina R. R. *Izv. Samar. Nauch. Centra RAN*, 2014, no. 1 (6), pp. 1689–1694 (in Russian).

16. **About** implementation of regional state ecological supervision in the territory of the Republic of Bashkortostan in 2014: state doc, M-vo prirodnopol'zovaniya i ekologii Respubliki Bashkortostan, Ufa, 2015, 150 p. (in Russian).

А. В. Вишнеков, д-р техн. наук, проф., avishnekov@hse.ru,

Е. М. Иванова, канд. техн. наук, доц., emivanova@hse.ru,

К. Э. Басова, магистрант, kezubakina@edu.hse.ru,

Е. О. Ветелина, магистрант, eovetelina@edu.hse.ru,

Национальный исследовательский университет "Высшая школа экономики", Москва

Интерактивный учебно-исследовательский комплекс для моделирования процессов в вычислительных системах

Рассматриваются вопросы моделирования работы функциональных блоков вычислительной системы, требования к моделям с точки зрения решения образовательных и исследовательских задач. Предлагается структура моделирующего комплекса, способы визуализации компонентов архитектуры и принципов функционирования блоков вычислительной системы. Приводятся описания алгоритмов работы моделирующих программ.

Ключевые слова: моделирование, архитектура вычислительной системы, визуализации компонентов архитектуры, алгоритмы функционирования блоков

Введение

В соответствии с Программой "Цифровая экономика Российской Федерации" ключевой задачей является подготовка кадров для цифровой экономики, развития прорывных информационных технологий [1]. Решение этой задачи во многом связано с повышением качества подготовки специалистов инженерных специальностей по направлению "Информатика и вычислительная техника". Важнейшим аспектом подготовки специалистов по информационным технологиям является изучение архитектуры современных компьютеров, принципов и алгоритмов функционирования блоков вычислительных систем. Сложность данной задачи заключается в следующем.

Во-первых, предметом изучения в данном случае является такой сложный технический объект, как вычислительная система. В настоящее время существует множество теоретических работ и учебных пособий, раскрывающих принципы построения архитектуры вычислительных систем (ВС) и алгоритмов функционирования их отдельных модулей. Однако учащимся сложно воспринимать большие объемы текстовой информации и на ее основе выстраивать представление об архитектурных особенностях той или иной части ВС и про-

цессе обработки данных в системе. В данном случае компьютерная модель устройства/алгоритма работы будет незаменима. Визуализация сделает любые концепции организации блоков ВС более наглядными, понятными, упростит понимание их работы. Это позволит более эффективно использовать аудиторные занятия преподавателем, существенно сократить время на дополнительную самостоятельную проработку материала учащимися, а также даст возможность обучения в режиме on-line по технологиям e-learning [2] или blended learning [3, 4], сократив требуемое время и средства для обсуждения и закрепления изучаемого материала.

Во-вторых, для комплексного изучения архитектуры ВС недостаточно теоретических сведений, необходимо использовать наглядные средства, демонстрирующие различные архитектурные принципы и технологии, применяемые при проектировании ВС. Невозможно проследить этапы обработки данных и особенности обработки данных в динамике на реальном компьютере. Многие особенности построения блоков вычислительной системы лежат на микроархитектурном уровне. В этом случае помочь в изучении архитектуры ВС могут компьютерные модели.

В-третьих, в отличие от аналитики на основе таблиц, рисунков или графиков модели-

рование дает понимание динамики процесса, позволяет наблюдать поведение реальной вычислительной системы во времени с необходимым уровнем детализации.

В четвертых, при компьютерном моделировании создается безрисковая среда, позволяющая безопасно и многократно применять и анализировать возможные сценарии декомпозиции компьютера и настройки его составных частей на конкретные режимы работы, которые выполнить в реальности невозможно.

Архитектура ВС, включая функциональную и структурную организацию, — это целый комплекс понятий, принципов взаимодействия программного обеспечения и аппаратной части ВС. Моделировать работу отдельных частей, входящих в понятие "архитектура ВС", аппаратных средств ВС, отдельных алгоритмов их функционирования можно инструментами САПР на языках поведенческого описания устройств (например, SystemVerilog, SystemC) с высокой точностью и степенью детализации модели. Однако, во-первых, подобные средства и модели, как правило, не являются свободно распространяемыми, и, во-вторых, эти средства предназначены для инженеров-проектировщиков ВС, а не для учебных целей, где такая степень детализации не требуется.

Предлагается создать моделирующий комплекс, в структуру которого входят блоки компьютерного моделирования наиболее сложных для понимания устройств/алгоритмов/протоколов функционирования ВС:

- конвейерной обработки потока машинных инструкций;
- способов уменьшения простоев конвейера;
- пошаговой работы ядра суперскалярного процессора;
- протоколов кэш-согласования в многоядерных системах;
- алгоритмов быстрого преобразования адресов с помощью Translation Lookaside Buffer (TLB);
- различных технологий многопоточной обработки данных;
- многоуровневой кэш-памяти;
- алгоритма отображения строк оперативной памяти на строки кэш-памяти;
- алгоритмов замещения строк кэш-памяти и др.

Постановка задачи

Задача заключается в создании моделирующего комплекса, позволяющего исследовать и изучать архитектурные особенности функцио-

нальных модулей вычислительной системы. Построить общую компьютерную модель вычислительной системы не представляется возможным в связи с высокой сложностью объекта моделирования. Поэтому целесообразно создать отдельные имитационные модели частей ВС с учетом технологических решений и архитектурных принципов, положенных в основу структуры того или иного блока ВС.

Исходя из основных архитектурных составляющих ядра ВС (процессор и память) следует рассмотреть модели именно этих частей ВС и их особенности. Основные требования, предъявляемые к моделям блоков ВС для решения учебных и исследовательских задач, следующие:

- степень детализации должна позволять решать учебные задачи и проводить исследование модели в целях оценки ресурсно-временных характеристик устройств;
- требуется простая и наглядная демонстрация алгоритмов работы исследуемого блока ВС;
- необходим дружественный интерфейс с контекстной помощью и/или подробное описание шагов моделирования;
- интерактивность модели должна давать возможность пошагового отслеживания поведения блока ВС или изменений состояния модели в динамике;
- должна быть возможность организации режима самостоятельного/удаленного обучения по технологиям e-learning или blended learning;
- необходима вариативность параметров модели для исследования поведения блоков ВС при различных начальных условиях, а также для обеспечения большого числа различающихся заданий для выполнения каждым студентом собственного варианта;
- время моделирования должно позволить изучить специфику процессов в отведенное по учебному графику время (как правило, два академических часа);
- необходима сопроводительная документация, включающая детальное описание изучаемого блока/алгоритма/принципа.

Описание метода решения задачи

Исходя из основных архитектурных составляющих ядра ВС (процессор и память) следует рассмотреть модели именно этих частей ВС. Рассмотрим особенности современных процессоров. Процессор — это сложное устройство, выполняющее вычисления по заданной пользователем программе и управля-

ющее ходом этих вычислений. Основными частями каждого процессора являются: ядро/ядра, внутренняя кэш-память, блоки управления обменом информацией между процессором и другими устройствами ВС. Поэтому представляется правильным создать модели работы процессорного ядра, модели работы кэш-памяти, модели управления обменом данными процессора и ряда других блоков, таких как оперативная память (ОП). Необходимо разработать модель центрального процессора (CPU) как универсального процессора наиболее общего типа, встречающегося в каждом современном компьютере.

Ядро современного CPU — это достаточно сложная система, и для его моделирования требуется создать комплекс моделей.

Модель 1. Основным механизмом, который в той или иной мере реализуется в каждом процессорном ядре для ускорения выполнения команд программы — это конвейерная обработка [5]. Поэтому первая модель, с исследования которой следует начать изучение процессора, будет "Модель конвейерной обработки потока машинных инструкций ядром центрального процессора". Модель "Конвейер ЦП" должна продемонстрировать:

- последовательность действий для исполнения машинной инструкции;
- разбиение машинной инструкции на несколько фаз (для их возможного совмещения для разных инструкций);
- разбиение CPU на несколько ступеней конвейера, исполняющих свою фазу инструкции;
- параллельное во времени исполнение разных фаз разных инструкций;
- потактовую загрузку ступеней конвейера CPU.

Модель 2. Поскольку конвейер команд является достаточно сложным функциональным блоком, то его модель также может получиться весьма сложной для понимания. Однако можно создать для исследования несколько взаимодополняющих моделей конвейера. Таким образом, модель "Конвейера ЦП" (или модификация модели 1) могла бы позволить проследить влияние параметров блоков процессора (ступеней конвейера) на скорость вычислений и загрузку исполнительных блоков и должна продемонстрировать:

- особенности программного кода, вызывающие задержки в вычислениях и простои ступеней конвейера [6, 7];
- способы влияния на ускорение вычислений путем изменения параметров конвейера инструкций;

- возможность накопления статистических данных для сравнения различных технологий и их влияния на ускорение вычислений.

Модель 3. Дальнейшее углубление в исследование принципов работы ядра CPU позволит понять и оценить сложность, необходимость и механизмы реализации принципов суперскалярности в современных процессорах [8—10], сочетающих особенности CISC- и RISC-архитектур и преимущества обоих подходов. Здесь существуют огромные перспективы для исследований. Предлагается остановиться на моделировании базовых механизмов [11] "Пошаговой работы ядра суперскалярного процессора", а именно:

- многофункциональной обработки данных с дублированием и переименованием переменных (регистров и ячеек памяти) [12];
- внеочередного исполнения операций [7, 8];
- предсказания переходов [13, 14] и спекулятивного исполнения программного потока [15].

Модель "Суперскаляр" должна продемонстрировать:

- все перечисленные выше базовые механизмы;
- принцип разбиения процессорного ядра на "front-end"- и "back-end"-кластеры [16—19];
- принцип преобразования потока сложных CISC-инструкций в простые RISC-подобные микрооперации [8, 10, 11, 15];
- ускоренную выборку инструкций из кэш-памяти уровня L1 [20, 21];
- исключение повторной загрузки данных из кэш-памяти уровня L1 [8];
- временную диаграмму загрузки блоков ядра процессора "front-end"- и "back-end"-кластеров;
- конвейерное исполнение инструкций и микроопераций;
- выполнение программы по верному и неверному предсказанию ветви с очисткой конвейера в последнем случае [9];
- возможность отслеживания влияния параметров исполнительного кластера ядра CPU и параметров программного кода на производительность.

Модель 4. Основная технология, позволяющая распараллелить вычисления — технология многопоточной обработки программных потоков ядрами CPU [22, 23]. Поэтому следующей моделью для исследования механизма вычислений, выполняемых процессором, будет "Моделирование различных технологий многопоточной обработки данных". Модель "Виды многопоточности" должна продемонстрировать:

- все способы многопоточной обработки программ;
- пошаговую работу процессора;
- процессорное время, затраченное при каждом способе;
- возможность сравнения всех способов.

Модель 5. Современная вычислительная система имеет сложную иерархически организованную подсистему памяти [5]: самая быстрая сверхоперативная память — внутренняя память процессора, состоящая из регистров/буферов процессорных ядер, кэш-памяти нескольких уровней (L0-L1-L2-L3-L4) [11, 24, 25], оперативной памяти, внешней памяти, реализованной на нескольких разнородных запоминающих устройствах (ЗУ), облачных хранилищах. Модель "Иерархия памяти" должна продемонстрировать:

- место каждого ЗУ в иерархии памяти ВС;
- назначение каждого типа памяти;
- объемы передаваемых данных за однократное обращение к ЗУ;
- латентность и скоростные характеристики всех типов ЗУ;
- способы доступа к данным для каждого типа памяти.

Модель 6. Иерархическая организация подсистемы памяти ВС привела к наличию нескольких типов адресов и адресных пространств. В результате при каждом обращении в память (за инструкциями программы или за данными) требуется задействовать блоки преобразования адресов и служебные таблицы. Таким образом, процесс вычисления конечного физического адреса оперативной памяти может быть достаточно сложной процедурой. Для ускорения переадресации в процессоре существуют блоки TLB [26]. Механизм их работы отражен в следующей модели "Алгоритмов быстрого преобразования адресов с помощью Translation Lookaside Buffer (TLB)". Модель "TLB" должна продемонстрировать:

- назначение и структуру TLB;
- разные случаи текущего состояния памяти (TLB, таблицы страниц);
- пошаговое вычисление физического адреса в разных случаях;
- преимущество использования TLB.

Модель 7. Многоядерные многопоточные процессоры могут выполнять программы, работающие с одними и теми же разделяемыми данными. Одновременное выполнение операций может привести к нарушению когерентности памяти [27]. Существуют протоколы, реализуемые аппаратно кэш-контроллером, которые по-

зволяют избежать неверного изменения общих данных. Реализуются они на последнем уровне иерархии кэш-памяти и демонстрируют различные технологии организации этого уровня. Поэтому следующая модель для исследования механизма вычислений, выполняемых процессором, — это "Моделирование протоколов кэш-согласования в многопроцессорных (многоядерных) системах". Модель "Кэш-протоколы" должна продемонстрировать:

- пошаговую работу каждого из исследуемых протоколов;
- изменения, проводимые в памяти (кэш и ОП);
- последовательность выполняемых действий при считывании, записи и удалении данных из кэш-памяти/ОП;
- возможность сравнения всех протоколов.

Модель 8. Кэш-память процессора строится на ассоциативных запоминающих устройствах, которые слишком сложны и дороги, поэтому существуют разные способы организации размещения (отображения) данных в кэш-памяти, различающихся степенью ассоциативности [11]. Кроме того, при поиске данных в кэш-памяти существует вероятность, что искомые данные там могут быть обнаружены (кэш-попадание) или не обнаружены (кэш-промах). Модель "Отображение строк в кэш-памяти" должна отражать:

- принцип организации кэш-памяти с разной степенью ассоциативности;
- последовательность действий при чтении/записи данных при кэш-попадании и кэш-промахе;
- возможность исследования влияния типа и размера памяти на число кэш-промахов.

Модель 9. Для наращивания емкости используется многоуровневая кэш-память (L1, L2, ...). Существуют различные стратегии организации управления обменом данными между уровнями кэш-памяти. Две наиболее различающиеся: инклюзивная и эксклюзивная. Модель "Многоуровневая кэш-память" должна позволить исследовать:

- влияние параметров кэш-памяти (емкости и латентности каждого уровня) на эффективность кэш-памяти в целом;
- сравнить каждый тип памяти.

Результаты работы моделирующего комплекса

Компьютерная модель 1 "Конвейер ЦП" отображает принцип конвейерной обработки инструкций программного кода для упро-

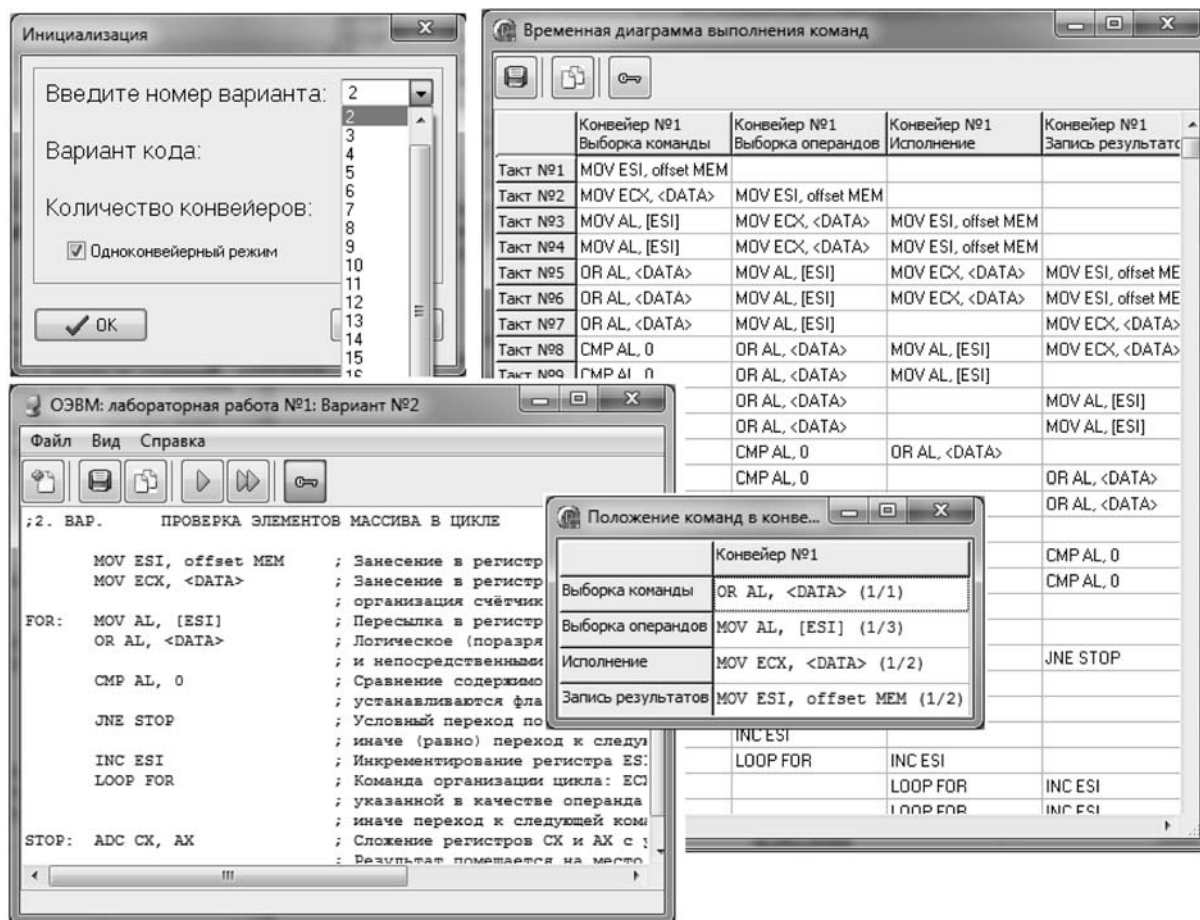


Рис. 1. Интерфейс компьютерной модели конвейера инструкций

ценного варианта процессора с упрощенным конвейером команд, состоящего из четырех ступеней. Моделирующая программа имеет несколько окон (рис. 1):

- главное меню — доступны функции начальных установок, запуска моделирования (по шагам и до конца), сохранения результатов, управления положением окон;
- инициализация — позволяет выбрать вариант модели и число конвейеров в процессоре (один — несколько);
- положение команд в конвейере — показывает при пошаговом моделировании положение команд на ступенях конвейера (в блоках процессора) на каждом такте моделирования выполнения фрагмента программного кода;
- временная диаграмма выполнения команд — отражает полную загрузку ступеней конвейера.

Выполняя моделирование, студенты могут проследить последовательность выполнения инструкций в процессоре, увидеть параллельное выполнение разных фаз разных инструкций блоками CPU, исследовать влияние типов исполняемых инструкций и числа конвейеров

на скорость вычислений. Моделирование помогает понять сложность механизма совмещения фаз и увидеть простые блоки CPU, вызванные разной длительностью фаз, зависимостью инструкций по данным или передачей управления. Для облегчения усвоения материала по теме разработано учебное пособие [28].

Модифицированная усложненная компьютерная модель 2 "Конвейера ЦП" (рис. 2) на вкладке "временная диаграмма" отражает аналогичную загрузку блоков процессора (ступеней конвейера), но позволяет исследовать влияние изменений исходных условий моделирования (или сложности организации процессора) на производительность, на загруженность блоков процессора и длительность простоев.

Следующие настройки допустимы:

- отмены повторного чтения операндов;
- уравнивания длительности всех фаз инструкции;
- изменение длины конвейера (числа ступеней);
- дублирования сильно загруженных ступеней конвейера;
- добавление блока предсказания переходов.

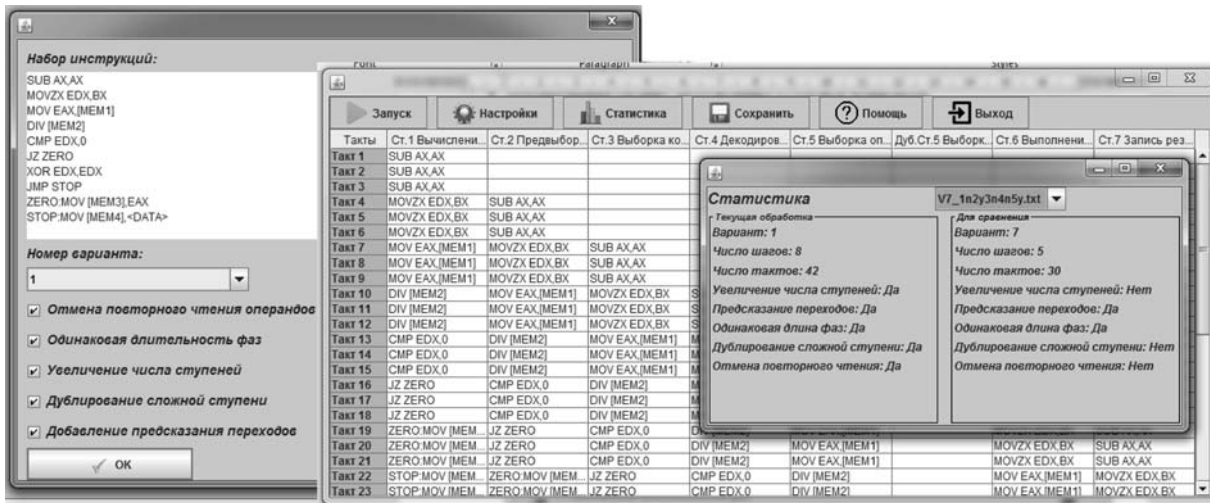


Рис. 2. Интерфейс усложненной компьютерной модели конвейера инструкций

Кроме того, можно проанализировать статистику — сравнительные данные по нескольким проходам программы с разными настройками.

Более сложная модель суперскалярного ядра CPU позволяет изучать и исследовать особенности микроархитектуры ядра современных процессоров Intel, такие как Skylake [8, 29], Zen [8, 30]. Она позволяет выполнить три исследова-

тельные работы и имеет многооконный интерфейс (рис. 3).

В окне "Задача" можно задать параметры процессора и программного кода и проследить влияние изменения этих параметров на загрузку блоков процессора и скорость вычислений. Окно "МОПы" демонстрирует инструкции из моделируемого фрагмента кода и результат их трансляции в микрооперации (мопы) с уче-

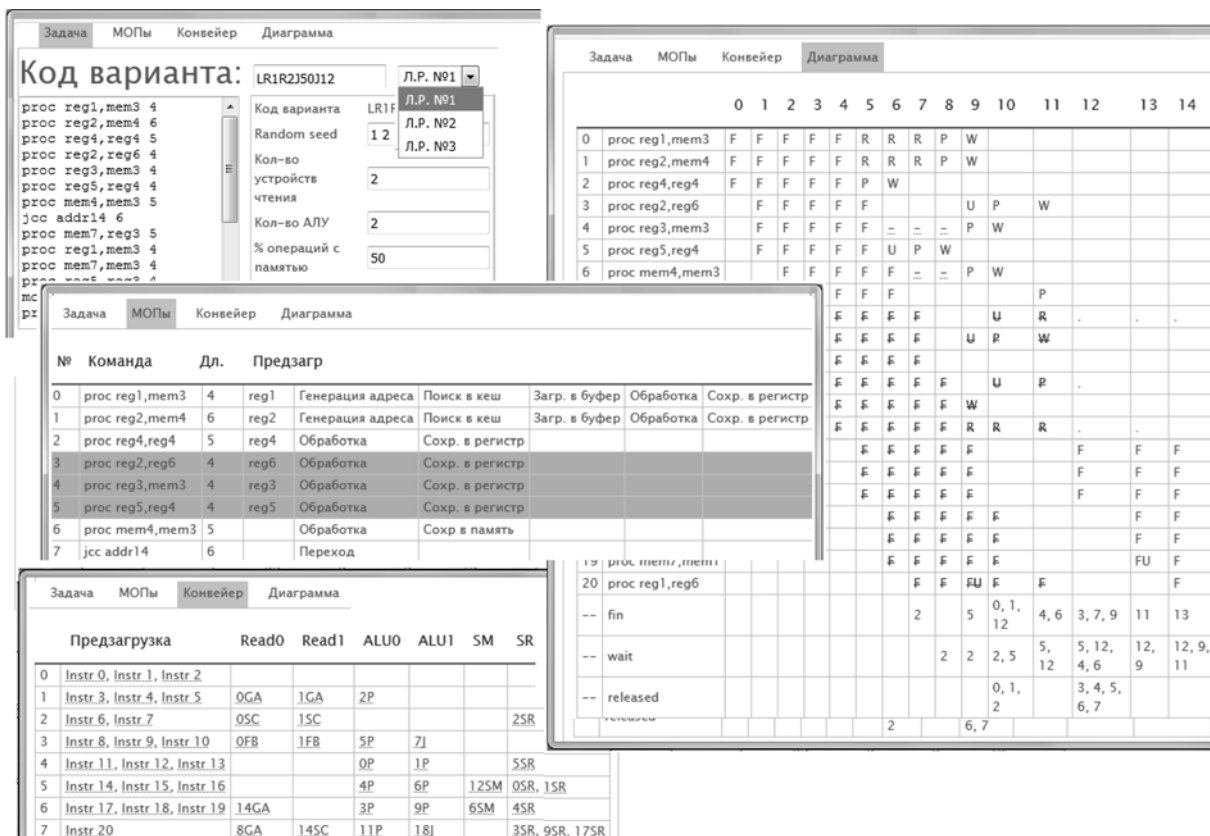


Рис. 3. Окна моделирующей программы "Суперскаляр"

том буферизации данных из ОП/КЭШ. Окно "Конвейер" позволяет увидеть выборку и декодирование набора инструкций во front-end-кластере ядра процессора и проследить динамику загрузки блоков back-end-кластера. Окно "Диаграмма" отражает последовательность выполнения инструкций по тактам и наглядно демонстрирует принцип внеочередного (Out-of-Order) исполнения операций. Также можно увидеть спекулятивное исполнение ветки программы с использованием механизма предсказания переходов (Branch Predict) с моделированием ситуации неверного предсказания адреса перехода и очистки конвейера. Микроархитектурные особенности ядра процессора и принципы его функционирования отражены

в учебном пособии [31], где также поясняется работа моделирующей программы.

Модель "Виды многопоточности" наглядно демонстрирует принцип исполнения нескольких программных потоков в процессоре одним из выбранных способов [22, 23]: CMT (course-gained multithreading), FMT (fine-gained multithreading), SMT (simultaneous multithreading) и дает возможность их сравнения (рис. 4).

Для заданного варианта кодов можно проследить переключения между потоками и сравнить процессорное время, затраченное при каждом способе.

Модель "TLB" (рис. 5) демонстрирует алгоритм трансляции виртуального адреса в физический с применением буфера ассоциативной

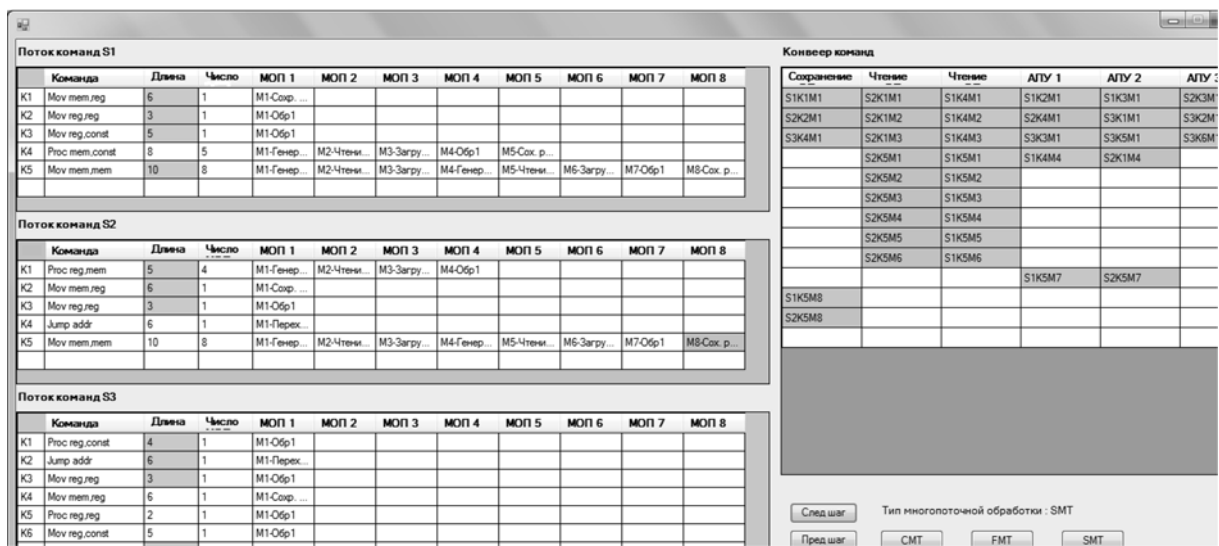


Рис. 4. Окна моделирующей программы "Суперкаляр"

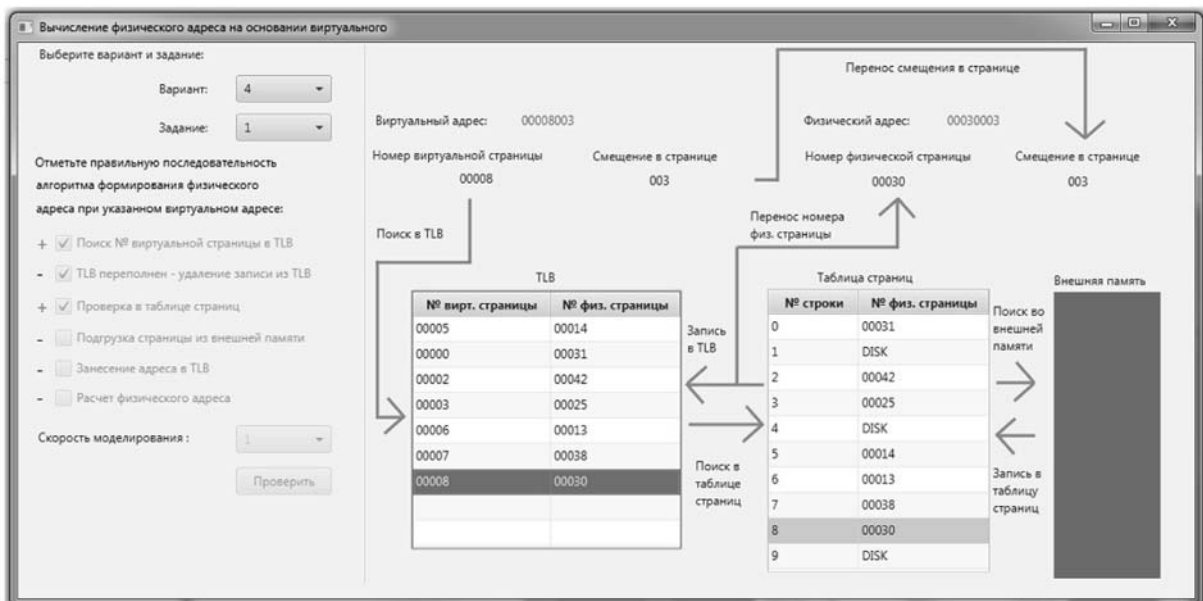


Рис. 5. Окно моделирующей программы после моделирования

трансляции и таблицы страниц. Модель позволяет проанализировать последовательность действий при различных начальных условиях:

- запись присутствует в TLB;
- запись отсутствует в TLB, но есть в таблице страниц;
- запись отсутствует в TLB и в таблице страниц.

Особенности механизма трансляции адресов и работы моделирующей программы отражены в учебном пособии [32].

Модель "Протоколы кэш-согласования" позволяет изучить способ обратной записи (write back) и сравнить несколько протоколов поддержания когерентности кэш-памяти последнего уровня иерархии LLC: MSI, MOSI, MESI, MOESI/MESIIF. Интерфейс программы включает несколько окон для отражения строк оперативной памяти и локальной кэш-памяти каждого из четырех процессоров/ядер, а также статусов каждой копии разделяемых данных в кэше (рис. 6). На каждом шаге моделирования в журнале выводится информация о выполненных действиях при реализации того или иного протокола. В результате моделирования можно исследовать скорость работы того или иного протокола (число выполняемых операций). Подробная информация о механизмах поддержания когерентности памяти и руководство по работе с моделирующей программой содержится в работе [33].

Модели алгоритмов замещения строк кэш-памяти (аналогично и для страниц ОП) позволяют проанализировать механизм работы запоминающего устройства, понять причины кэш-промахов, проследить проверку важности строк кэш-памяти, изменения данных и тегов (ссылок на записи) для небольшого числа последовательности обращений в память (рис. 7).

Рассмотрены все основные алгоритмы замещения: RND, LFU, LRU, MRU, FIFO и гибридный алгоритм ARC [34]. Модель демонстрирует для каждого алгоритма особенности его реализации: содержимое строк ОП, состояние кэш (памяти тегов и памяти данных), состояние стека протоколов (временные метки или счетчик обращений).

Модель "Отображение строк в кэш-памяти" позволяет исследовать различные механизмы отображения строк ОП на строки кэш-памяти. Рассмотрены три варианта отображения: прямое, полностью ассоциативное, наборно-ассоциативное (рис. 8). На каждом шаге моделирования можно проанализировать порядок работы блоков кэш-контроллера, последовательность поиска данных с указанным адресом, результаты поиска: кэш-попадание или кэш-промах. В качестве исходных параметров задается число наборов/строк в кэш-памяти, изменяя которые можно исследовать их влияние на эффективность кэш-памяти.

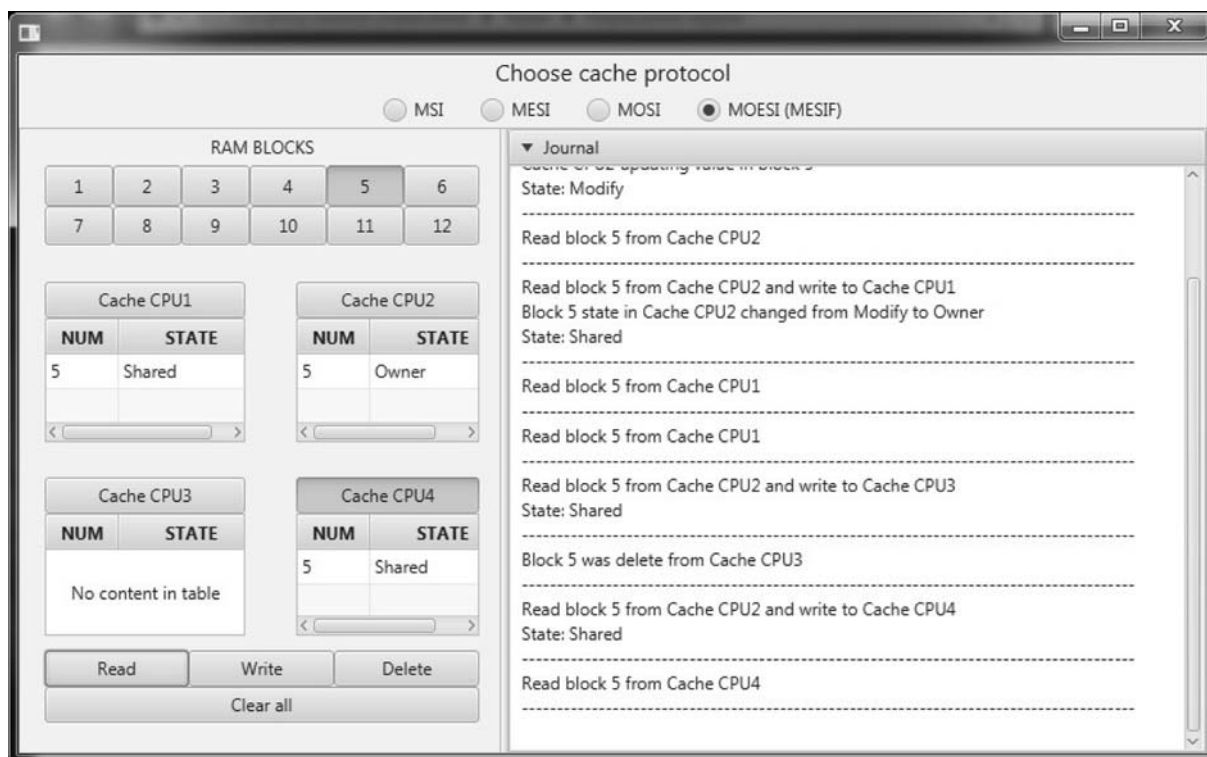


Рис. 6. Интерфейс модели "Протоколы кэш-согласования"

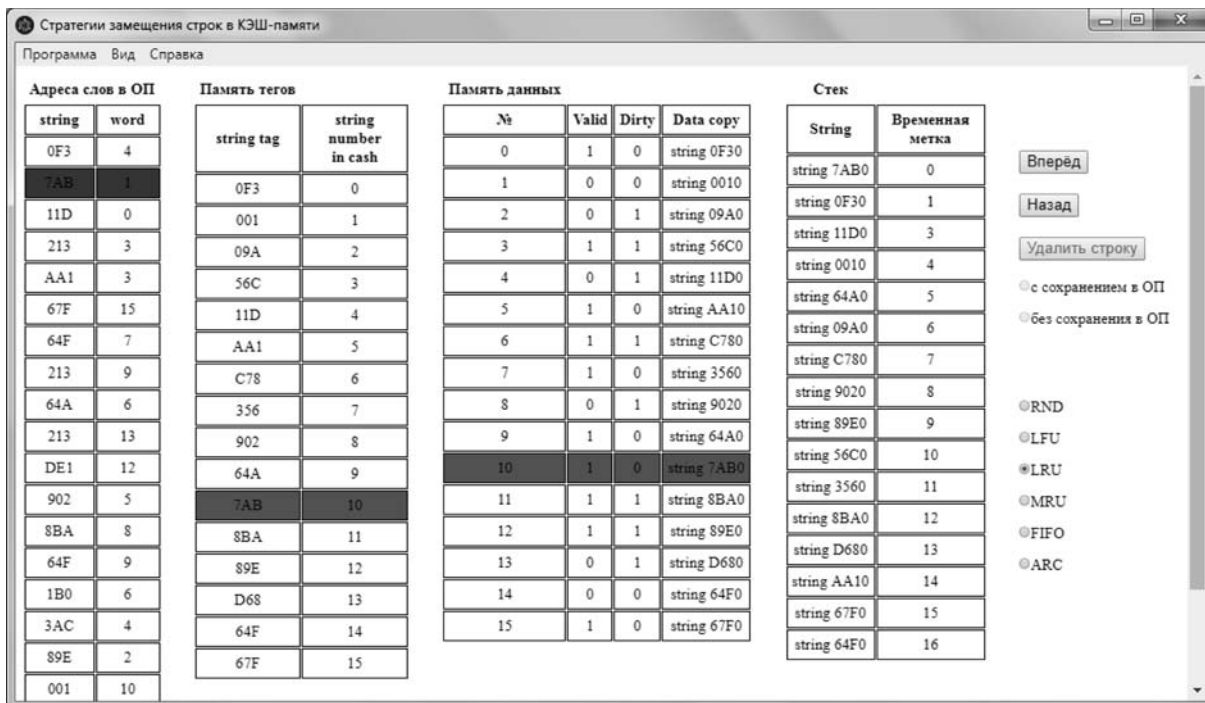


Рис. 7. Интерфейс модели "Алгоритмы замещения строк кэш-памяти"

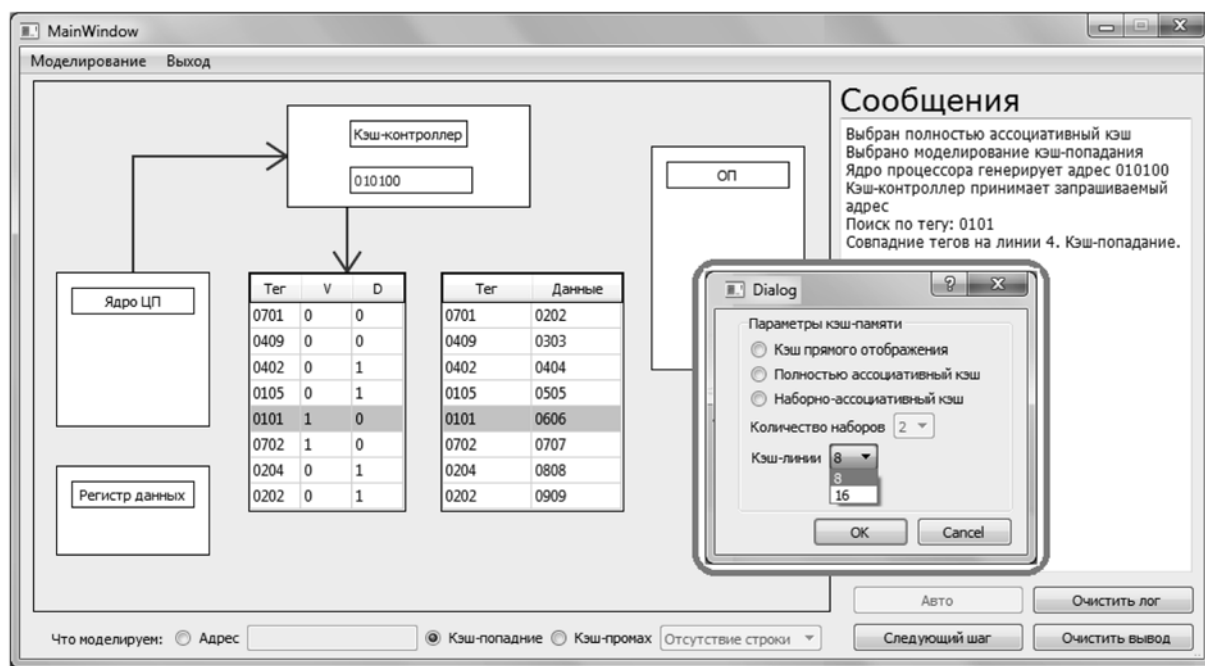


Рис. 8. Интерфейс модели "Отображение строк в кэш-памяти"

Модель "Многоуровневая кэш-память" позволяет сравнить два типа организации многоуровневой кэш-памяти: инклюзивный и эксклюзивный [35]. Меняя параметры моделирования, можно проследить скорость заполнения и процент использования каждого уровня кэш-памяти в обоих случаях, влияние типа кэш-памяти, его размера и числа строк на эф-

фективность кэш-памяти: латентность операций чтения и на число кэш-промахов (рис. 9).

Заключение

Моделирующий комплекс включает необходимое число моделей для изучения архитектуры

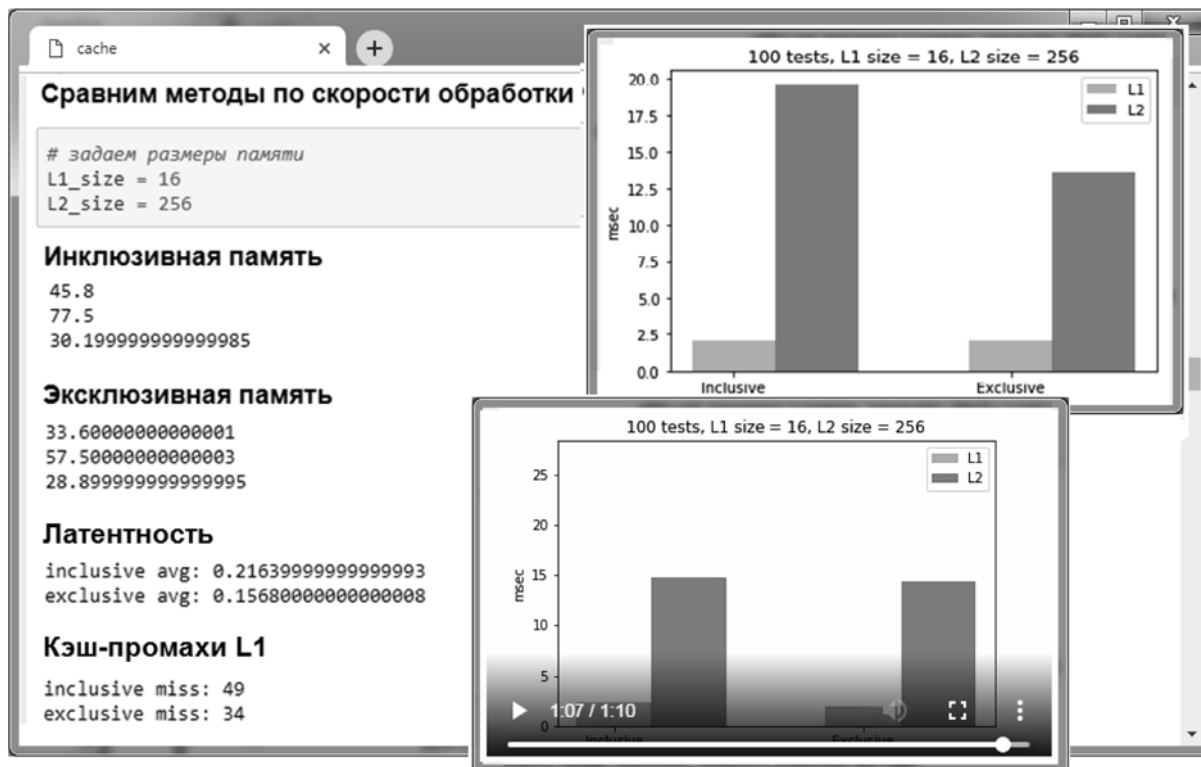


Рис. 9. Интерфейс модели "Многоуровневая кэш-память"

вычислительной системы, позволяет исследовать особенности алгоритмов функционирования отдельных блоков, узлов ВС. Разработанные имитационные модели блоков ВС обладают необходимой степенью абстракции и отвечают критериям наглядности, удобства использования, интерактивности, вариативности параметров. Предлагаемый моделирующий комплекс позволяет педагогам выстроить процесс обучения, эффективно сочетающий в себе элементы очного и on-line образования.

Важным достоинством предлагаемого моделирующего комплекса является возможность вариации параметров моделей отдельных блоков ВС, что дает возможность исследования ресурсно-временных характеристик процессов обработки данных блоками вычислительной системы.

Список литературы

1. **Национальная** программа "Цифровая экономика Российской Федерации". URL: <http://government.ru/rugovclassifier/614/events/> (дата обращения 29.03.19).
2. **Association** of e-Learning specialists. URL: <http://www.elearningnc.gov> (дата обращения 15.03.19).
3. **Garrison D. R., Vaughan N. D.** Blended learning in higher education: Framework, principles, and guidelines. San Francisco, CA: Jossey-Bass, 2008.
4. **Waha B., Davis K.** University students' perspective on blended learning // Journal of Higher Education Policy and Management. 2014. Vol. 36, N. 2. P. 172–182. DOI: 10.1080/1360080X.2014.884677.

5. **Таненбаум Э.** Архитектура компьютера. СПб.: Питер, 2014. 811 с.

6. **Kanter D.** Intel's Sandy Bridge Microarchitecture. Website Real World Tech. 2010. URL: <https://www.realworldtech.com/sandy-bridge/10/> (дата обращения 25.02.19).

7. **Intel** Sandy Bridge microarchitecture overview. Out-of-Order Cluster. URL: <https://hw-lab.com/intel-sandy-bridge-cpu-microarchitecture.html/4> (дата обращения 13.01.19).

8. **Intel® 64 and IA-32 Architectures Optimization Reference Manual.** 2018. URL: <https://docsliade.net/download/link/intel-64-and-ia-32-architectures-optimization-reference-64-and-ia-32-architectures> (дата обращения 10.01.19).

9. **Корогод С.** Новое в архитектуре процессоров Silvermont Intel Atom Z3xxx. 2013. URL: <http://www.ixbt.com/portopc/atom-clover5.shtml> (дата обращения 28.03.19).

10. **Процессорная** микроархитектура AMD Bulldozer. URL: <http://compress.ru/article.aspx?id=21786&iid=995> (дата обращения 15.03.19).

11. **Электронная** энциклопедия процессорных терминов. URL: <http://www.ixbt.com/cpu/cpu-pedia.shtml> (дата обращения 15.03.19).

12. **Maleen A., Suvinay S., Mark J. C., Joel E., Daniel S.** SAM: Optimizing Multithreaded Cores for Speculative Parallelism // 26th International Conference "Parallel Architectures and Compilation Techniques (PACT)". 2017 P. 64–78.

13. **Coursera.** Branch Prediction Introduction. URL: <https://www.coursera.org/lecture/comparch/branch-prediction-introduction-BF58C> (дата обращения 29.03.19).

14. **Branch** Prediction Review. URL: <https://courses.cs.washington.edu/courses/csep548/06au/lectures/branchPred.pdf> (дата обращения 10.02.19).

15. **Intel** Analysis of Speculative Execution Side Channels. White Paper. 2018. URL: <https://newsroom.intel.com/wp-content/uploads/sites/11/2018/01/Intel-Analysis-of-Speculative-Execution-Side-Channels.pdf> (дата обращения 29.03.19).

16. **Hinton G., Sager D., Upton M.** etc. The Microarchitecture of the Pentium4 Processor. Intel Technology Journal Q1, 2001. URL: <http://www.ecs.umass.edu/ece/koren/ece568/papers/Pentium4.pdf> (дата обращения 29.03.19).
17. **Swinburne R.** Feature-Intel Core i7-Nehalem Architecture Dive. bit-tech. 2008. URL: <http://www.bit-tech.net/reviews/tech/cpus/intel-core-i7-nehalem-architecture-dive/5/> (дата обращения 28.03.19).
18. **Озеров С.** Архитектура CPU. Computerra. 2005, 37(609). URL: <http://www.kinnet.ru/cterra/609/233266.html> (дата обращения 28.03.19).
19. **Нечай О.** Подборка статей автора. URL: http://www.libma.ru/компьютеры_и_интернет/cifrovoi_zhurnal_компьютера_74/p2.php (дата обращения 28.03.19).
20. **Полуялов А.** В ожидании Willamette — история архитектуры IA-32 и как работают процессоры семейства P6. 2000. URL: <https://www.ixbt.com/cpu/pentium4-1.html> (дата обращения 28.03.19).
21. **Гарматюк С.** "Новая старая" архитектура Core i7: чего больше — сходства или различий? 2008. URL: <https://www.ixbt.com/cpu/intel-ci7-theory.shtml> (дата обращения 28.03.19).
22. **Архитектурные аспекты параллелизма.** URL: <http://www.intuit.ru/studies/courses/4447/983/lecture/14921> (дата обращения 15.03.19).
23. **Многонитевая архитектура микропроцессоров.** Открытые системы. URL: <https://www.osp.ru/os/2002/01/180943/> (дата обращения 15.03.19).
24. **Intel's Embedded DRAM: New Era of Cache Memory.** URL: https://www.eetimes.com/author.asp?section_id=36&doc_id=1323410 (дата обращения 28.03.19).
25. **Процессоры Intel Sandy Bridge — все секреты.** URL: <https://www.ixbt.com/cpu/sandy-bridge.shtml> (дата обращения 28.03.19).
26. **Intel 64 and IA-32 Architectures Software Developer's Manual.** URL: <http://www.intel.ie/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf> (дата обращения 28.03.19).
27. **Neupane, Mahesh.** Cache Coherence. URL: https://web.archive.org/web/20100620091706/http://cse.csusb.edu/schubert/tutorials/csc610/w04/MN_Cache_Coherence.pdf (дата обращения 01.03.19).
28. **Иванова Е. М.** Моделирование работы многоконвейерного процессора. М.: МИЭМ, 2011.
29. **Skylake (client) — Microarchitectures — Intel.** URL: [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(client\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client)) (дата обращения 18.03.19).
30. **Zen — Microarchitectures — AMD.** URL: <https://en.wikichip.org/wiki/amd/microarchitectures/zen> (дата обращения 21.03.19).
31. **Иванова Е. М.** Модель работы ядра суперскалярного ЦП. М.: НИУ ВШЭ, 2018.
32. **Иванова Е. М.** Быстрое преобразование адресов с помощью Translation Lookaside Buffer (TLB). М.: НИУ ВШЭ, 2018.
33. **Иванова Е. М.** Модель протоколов КЭШ-согласования в многопроцессорных (многоядерных) системах. М.: НИУ ВШЭ, 2018.
34. **Петраков И. Е.** Модели и алгоритмы гибридизации стратегий кэширования // Электронный научный журнал "Программные продукты, системы и алгоритмы". 2016. № 2. DOI: 10.15827/2311-6749.19.187.
35. **Гарматюк С.** Современные десктопные процессоры архитектуры x86: общие принципы работы. URL: <https://www.ixbt.com/cpu/x86-cpu-faq-2006.shtml> (дата обращения 29.03.19).

A. V. Vishnekov, Ph. D., Professor, e-mail: avishnekov@hse.ru,

Е. М. Иванова, Ph. D., Associate Professor, e-mail: emivanova@hse.ru,

К. Е. Basova, master's student, "Computer systems and networks", e-mail: kezubakina@edu.hse.ru,

Е. О. Vetelina, master's student, "Computer systems and networks", e-mail: eovetelina@edu.hse.ru,

National Research University Higher School of Economics, Moscow, 123458, Russian Federation

Interactive Educational-Research Complex for Processes Simulation in Computer Systems

This article aims at the improvement of the engineer training quality in the field of "computer Science and engineering" using on-line and blended learning technology. We consider studying of modern computers architecture as an example. The complexity of the modeling object, the multi-faceted process of studying the computer architecture, the limited period — are the factors that determine the difficulty of the problem. A computer simulator allows you to explore the stages and features of the calculation process in dynamics with the necessary detail. It can't be done any other way. Also, computer modeling creates a risk-free environment that allows you to safely and repeatedly apply and analyze possible scenarios, decomposition of computer nodes and configuring its components for different modes of operation. This is either impossible to perform on real devices or may result in computer failure under students' inept actions. The paper discusses the problems of the computer units and devices simulating, the requirements for models in terms of solving educational and research problems. We propose a structure of the simulating complex, visualization methods of computer architecture components and principles of computer system device's operation, algorithms descriptions of the modeling programs. Since the computer architecture implies a whole complex of concepts, including functional and structural organization, software and hardware interaction algorithms, the considered simulating complex consists of several simulators: step-by-step operation of the superscalar processor core; cache coherence protocols in multiprocessor (multi-core) systems; algorithms for fast address translation with the help of Translation Lookaside Buffer (TLB); various technologies for multithreaded data processing; pipeline processing of the machine instructions stream; memory paging algorithms; cache mapping algorithms; cache replacement algorithms; etc. The main functions and features of each model are considered. An important advantage of the proposed simulating complex is the ability to vary the parameters of models of specific computer devices, which makes it possible to explore the resource/time characteristics of data processing by computer system devices.

Keywords: computer architecture, simulating, visualization of the computer architecture components, algorithms of the computer devices operating

DOI: 10.17587/it.25.490-501

References

1. **National** program "Digital economy of the Russian Federation", available at: <http://government.ru/rugovclassifier/614/events/> (retrieved 29.03.19) (in Russian).
2. **Association** of e-Learning specialists, available at: <http://www.elearningnc.gov> (retrieved 15.03.19).
3. **Garrison D. R., Vaughan N. D.** Blended learning in higher education: Framework, principles, and guidelines, San Francisco, CA, Jossey-Bass, 2008.
4. **Waha B., Davis K.** University students' perspective on blended learning. *Journal of Higher Education Policy and Management*, vol. 36, 2014, no. 2, pp. 172–182. DOI: 10.1080/1360080X.2014.884677.
5. **Tanenbaum A. S., Austin T.** Structured Computer Organization, Prentice Hall, 2012.
6. **Kanter D.** Intel's Sandy Bridge Microarchitecture, available at: <https://www.realworldtech.com/sandy-bridge/10/> (retrieved 25.02.19).
7. **Intel** Sandy Bridge microarchitecture overview. Out-of-Order Cluster, available at: <https://hw-lab.com/intel-sandy-bridge-cpu-microarchitecture.html/4> (retrieved 13.01.19).
8. **Intel® 64 and IA-32 Architectures Optimization Reference Manual** (2018), available at: <https://docslide.net/download/link/intel-64-and-ia-32-architectures-optimization-reference-64-and-ia-32-architectures> (retrieved 10.01.19).
9. **Korogod S.** New in the Silvermont architecture of Intel Atom Z3xxx Processors, available at: <http://www.ixbt.com/portopc/atom-clover5.shtml> (retrieved 28.03.19) (in Russian).
10. **Pakhomov S.** AMD Bulldozer processor microarchitecture, available at: <http://compress.ru/article.aspx?id=21786&iid=995> (retrieved 15.03.19) (in Russian).
11. **Digital** encyclopedia of processor terms, available at: <http://www.ixbt.com/cpu/cpu-pedia.shtml> (retrieved 15.03.19) (in Russian).
12. **Maleen A., Suvinay S., Mark J. C., Joel E., Daniel S.** SAM: Optimizing Multithreaded Cores for Speculative Parallelism, *PACT 2009 – 26th International Conference on Parallel Architectures and Compilation Techniques*, Raleigh, North Carolina, USA, pp. 64–78.
13. **Coursera.** Branch Prediction Introduction, available at: <https://www.coursera.org/lecture/comparch/branch-prediction-introduction-BF58C> (retrieved 29.03.19).
14. **Control** Hazards. Branch Prediction Review, available at: <https://courses.cs.washington.edu/courses/csep548/06au/lectures/branchPred.pdf> (retrieved 10.02.19).
15. **Intel** Analysis of Speculative Execution Side Channels. White Paper. 2018, available at: <https://newsroom.intel.com/wp-content/uploads/sites/11/2018/01/Intel-Analysis-of-Speculative-Execution-Side-Channels.pdf> (retrieved 29.03.19).
16. **Hinton G., Sager D., Upton M., Boggs D., Carmean D., Kyker A., Roussel P.** The Microarchitecture of the Pentium4 Processor. *Intel Technology Journal Q1*, available at: <http://www.ecs.umass.edu/ece/koren/ece568/papers/Pentium4.pdf> (retrieved 29.03.19).
17. **Swinburne R.** Feature-Intel Core i7-Nehalem Architecture Dive. bit-tech, available at: <http://www.bit-tech.net/reviews/tech/cpus/intel-core-i7-nehalem-architecture-dive/5/> (retrieved 29.03.19)
18. **Ozerov S.** CPU Architecture. Computerra. 2005, 37(609), available at: <http://www.kinnet.ru/cterra/609/233266.html> (retrieved 28.03.19) (in Russian).
19. **Nechaj O.** Author's articles compilation, available at: http://www.libma.ru/kompyutery_i_internet/cifrovoy_zhurnal_kompyuterra_74/p2.php (retrieved 28.03.19) (in Russian).
20. **Poluyvalov A.** Waiting for the Willamette — the history of the IA-32 architecture and how do the P6 family processors work, available at: <https://www.ixbt.com/cpu/pentium4-1.html> (retrieved 28.03.19) (in Russian).
21. **Garmatyuk S.** "New old" core i7 architecture: what's more — similarities or differences?, available at: <https://www.ixbt.com/cpu/intel-ci7-theory.shtml> (retrieved 28.03.19) (in Russian).
22. **Architectural** aspects of concurrency, available at: <http://www.intuit.ru/studies/courses/4447/983/lecture/14921> (retrieved 15.03.19) (in Russian).
23. **Multithreaded** architecture of microprocessors. *Open systems*, available at: <https://www.osp.ru/os/2002/01/180943/> (retrieved 15.03.19) (in Russian).
24. **Intel's** Embedded DRAM: New Era of Cache Memory, available at: https://www.eetimes.com/author.asp?section_id=36&doc_id=1323410 (retrieved 28.03.19).
25. **Intel** Sandy Bridge processors — all secrets, available at: <https://www.ixbt.com/cpu/sandy-bridge.shtml> (retrieved 28.03.19) (in Russian).
26. **Intel** 64 and IA-32 Architectures Software Developer's Manual, available at: <http://www.intel.ie/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf> (retrieved 28.03.19).
27. **Neupane, Mahesh.** Cache Coherence, available at: https://web.archive.org/web/20100620091706/http://cse.csusb.edu/schubert/tutorials/csci610/w04/MN_Cache_Coherence.pdf (retrieved 01.03.19).
28. **Ivanova E. M.** Simulation of multi-pipeline processor operation, Moscow, MIEM, 2011 (in Russian).
29. **Skylake** (client) — Microarchitectures — Intel, available at: [https://en.wikichip.org/wiki/intel/microarchitectures/skylake_\(client\)](https://en.wikichip.org/wiki/intel/microarchitectures/skylake_(client)) (retrieved 18.03.19).
30. **Zen** — Microarchitectures — AMD, available at: <https://en.wikichip.org/wiki/amd/microarchitectures/zen> (retrieved 21.03.19).
31. **Ivanova E. M.** Model of superscalar CPU core operation, Moscow, NRU HSE, 2018 (in Russian).
32. **Ivanova E. M.** Fast address translation using Translation Lookaside Buffer (TLB), Moscow, NRU HSE, 2018 (in Russian).
33. **Ivanova E. M.** Model of cache coherence protocols in a multi-core systems, Moscow, NRU HSE, 2018 (in Russian).
34. **Petrakov I. E.** Models and algorithms of caching strategies hybridization. *Digital scientific journal "Software products, systems and algorithms"*, 2016, no. 2, DOI: 10.15827/2311-6749.19.187 (in Russian).
35. **Garmatyuk S.** Modern desktop processors of x86 architecture: General operation principles, available at: <https://www.ixbt.com/cpu/x86-cpu-faq-2006.shtml> (retrieved 29.03.19) (in Russian).

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКЕ И ПРОИЗВОДСТВЕ

INFORMATION TECHNOLOGY IN THE ECONOMY AND PRODUCTION

УДК 004.023

DOI: 10.17587/it.25.502-511

В. В. Грибова, д-р техн. наук, ст. науч. сотр.,
зам. директора по научной работе, e-mail: gribova@iacp.dvo.ru,
А. С. Величко, канд. физ.-мат. наук, доц., науч. сотр., e-mail: vandre@iacp.dvo.ru,
А. В. Колмогоров, аспирант, e-mail: sunrise3323@gmail.com,
Институт автоматизации и процессов управления Дальневосточного отделения
Российской академии наук, г. Владивосток

Концепция цифровой платформы для производственной деятельности*

Рассматривается задача организации цифрового производства. Проведен анализ существующих решений, выявлены их преимущества и недостатки. Описаны требования к реализации архитектуры цифровой платформы для организации основных этапов производства. Обоснованы задачи, решаемые каждым функциональным модулем, методы их реализации.

Ключевые слова: цифровая экономика, организация производства, производственно-транспортные задачи, информационная платформа, инструментальный комплекс

Введение

Цифровые технологии неуклонно меняют расстановку сил на глобальных рынках. Традиционные подходы к организации и управлению производством теряют свою актуальность, становясь неэффективными. Ответом на сложившуюся ситуацию стала утвержденная распоряжением Правительства РФ от 28 июля 2017 года программа "Цифровая экономика Российской Федерации". Одной из ключевых задач данной программы является формирование исследовательских компетенций и технологических заделов, направленных на создание цифровых платформ, обеспечивающих национальную безопасность и технологическую независимость России.

В условиях рыночной экономики важными условиями деятельности бизнеса в отраслях, характеризующихся высококонкурентной структурой рынка, является снижение всех видов издержек, увеличение своей рыночной

доли в отрасли и расширение географии поставок продукции.

Актуальной для развития бизнеса является оптимизация процессов производственной деятельности в целях выбора наиболее выгодных условий и уменьшения расходов на каждом этапе. С точки зрения региональной специфики деятельности бизнеса в российской экономике, например на Дальнем Востоке, проблема логистических (складских) и транспортных издержек стоит особо остро. Развитие цифровых технологий позволяет бизнесу управлять всеми этими процессами с помощью разрабатываемых и внедряемых цифровых платформ.

Основное внимание статьи сосредоточено на внедрении научных и информационных технологий в промышленность и в производство. Задачами данной работы являются анализ существующих решений, описание требований, концепции и архитектуры расширяемой цифровой платформы для организации цифрового производства, выбора поставщиков и заключения договоров в непрерывном производственном процессе.

* Работа выполнена при частичной финансовой поддержке РФФИ (проекты 18-07-01079, 19-07-00244).

1. Обзор существующих программных решений для цифровизации этапов и процессы деятельности фирмы

Можно выделить несколько основных подходов к организации производства, в том числе цифрового, и направлений производственной деятельности, на которых с точки зрения разработчиков должны базироваться создаваемые цифровые платформы или экосистемы. Так, IBM предлагает сосредоточиться на трех основных направлениях [1]: оптимизация производственной цепочки создания стоимости; получение максимальной отдачи от капитальных проектов; получение максимальной отдачи от промышленного оборудования. Далее для каждого из этих направлений предлагаются цифровые решения, в том числе для конкретных отраслей [2].

В работе [3] авторы считают, что успешное производство, прежде всего, зависит от характеристик и свойств продукта. Необходимо сосредоточиться на выборе продуктов для производства, подборе его технических характеристик, выборе материала, создании ассортимента. Авторы приходят к выводу, что постоянное совершенствование бизнеса является первоочередной задачей любой организации в условиях рыночной конкуренции, фирмам необходимо планировать, управлять и контролировать бизнес-процессы для удовлетворения потребности потребителей, выражаемой в форме рыночного спроса.

В работе [4] акцент сделан на вопросах организации производства в рамках таких направлений деятельности фирмы, как прогноз спроса на предполагаемый продукт производства, определение потенциальных вариантов производства, их анализ и выбор оптимального варианта в соответствии с заданными критериями, мониторинг и контроль процесса производства, гибкая корректировка производственных планов из-за изменений спроса на рынке.

Актуальной задачей цифровизации деятельности фирмы является создание новых цифровых платформ (экосистем), поддерживающих как различные направления производственной деятельности, так и различные методы их организации, поскольку "остро ощущается их нехватка, особенно в России" [5].

Таким образом, на основе вышеприведенного анализа источников можно выделить следующие основные направления и процессы производственной деятельности с учетом странственной организации работы фирмы:

I. Организация и планирование:

- определение одного или нескольких мест размещения производств;
- планирование технологического процесса и выбор необходимых ресурсов для осуществления производства (ресурсы);
- выбор оптимальных способов и мест приобретения ресурсов;

II. Осуществление производственной деятельности и управление ею:

- управление процессом поставок и закупок ресурсов и их оплатой;
- производственная деятельность в рамках технологического процесса;
- реализация готовой продукции и их оплата.

III. Анализ, мониторинг процессов деятельности, развитие бизнеса.

Проанализируем предлагаемые в настоящее время программные продукты, которые направлены на цифровизацию процессов деятельности фирмы. Для этого, например, в работе [6] предлагается выделить следующие типы программных инструментов, систем и платформ:

1) социальные сети: Facebook, Twitter, ВКонтакте, Instagram и др.;

2) поисковые системы: Google, Yandex, Yahoo и др.;

3) мессенджеры: Telegram, WhatsApp, Viber, Uber и др.;

4) специализированные программные системы и инструменты: 1С, SAP, Парус, Галактика;

5) универсальные платформы электронной торговли (коммерции): eBay, Amazon, Alibaba, Taobao, Aliexpress и др.

Основными информационными технологиями, поддерживающими функционирование процессов цифровизации деятельности фирмы, являются, прежде всего, интернет-технологии, облачные технологии, технологии искусственного интеллекта и больших данных, блокчейн-технологии, включая реализованные на ее основе так называемые умные контракты, и др. В зарубежных источниках присутствуют исследования, цель которых — продемонстрировать практическое использование новых технологий для цифровизации. Так, в работе [7] описывается подход использования умных контрактов для мира "Интернета вещей".

Поскольку универсальные платформы электронной торговли, специализированные программные системы и инструменты являются комплексными решениями, интегрирующими в себе различные информационные техноло-

гии и подходы для цифровизации деятельности фирмы, сделаем акцент на их анализе.

Системы электронной торговли (коммерции) можно разбить на два основных класса: платформы для разработки интернет-магазинов, например, Magento Community Edition, Shopify, PrestaShop, OpenCart, osCommerce и др. и платформы-агрегаторы, такие как Amazon, Alibaba, Яндекс Маркет, БЕРУ, объединяющие онлайн-сеть продавцов, маркетологов, поставщиков услуг, логистических компаний и производителей.

Основными функциями платформ для разработки интернет-магазинов являются: разработка индивидуального сайта магазина на основе предоставляемых шаблонов и плагинов для размещения и редактирования товаров, работа с платежными агрегаторами для продажи продукции, проведение маркетинговых компаний, формирование отчетов и статистики.

К недостаткам таких решений относятся: сложность настройки индивидуальных требований к сайту-магазину, зависимость от плагинов, плата за используемые облачные технологии, в том числе оплата платформы, на которой веб-ресурс размещен.

Платформы-агрегаторы имеют более широкие функциональные возможности. Они, прежде всего, выступают как сборщики сведений о ценах различных товаров отдельных производителей, а также позволяют крупным бизнесам и брендам напрямую торговать в интернете. Агрегаторы объединяют в единую систему продавцов и верифицируют каждого продавца на своей платформе. Платформы-агрегаторы выступают как посредники между продавцом и покупателем, обеспечивая безопасность операции обмена. Платформы-агрегаторы имеют единую логистическую систему, которая состоит из точек приема заказа и логистических компаний, расположенных во всех крупных городах, в которую поставщики сдают продукцию для отправки покупателю. На платформах-агрегаторах часто продавцы являются производителями, это уменьшает риск купить продукцию с дополнительной наценкой.

К недостаткам данного типа средств относится их направленность, прежде всего, на торговлю товарами, а не их производство. Рассматривая такие средства для производства, можно отметить, что их функциональные возможности сильно ограничены. Это делает возможным использовать их только как частное решение для этапа производства, направленного на поиск поставщика и логистики.

Типичным представителем платформ-агрегаторов является компания Alibaba и их сервисы [8], в частности, сервис Taobao, предоставляющий возможность поставщикам продавать оптом продукцию, которую можно использовать как производственный ресурс для новой продукции.

Еще одним примером является компания Amazon. В работе [9] авторы показывают, что тенденция развития цифровизации движется в сторону логистики. Так, в 2017 году Amazon добился значительного прогресса в создании своего подразделения по транспортировке и логистике. Для поддержки своих продаж ритейлер подписал договор аренды 40 грузовых самолетов с двумя американскими авиаперевозчиками Atlas Air Worldwide и Air Transport Services Group. Кроме того, Amazon провел первую успешную поставку через свою программу Prime Air drone в Великобритании.

Специализированные системы и инструменты поддерживают решение определенного круга задач и являются универсальными (не ориентированными на какую-либо отрасль производства) либо автоматизируют бизнес-процессы в конкретной отрасли. Одним из универсальных инструментов является платформа "1С Предприятие", которая включает комплекс программ автоматизации учета и управления предприятием [10]. К функциям учета относятся ведение бухгалтерского и налогового учета, расчет заработной платы, составление отчетной документации и т. д. Функции управления классифицируются по направлениям автоматизации и группам пользователей. Эти функции системы имеют своей целью обеспечение руководителей информацией, необходимой для оценки ситуации и принятия актуальных решений (механизмы бюджетирования, анализ рентабельности деятельности предприятия, сбыта продукции и др.) [11].

Другим универсальным программным инструментом, используемым за рубежом, является SAP. В статье [12] авторы характеризуют систему и ее возможности для управления внутренними процессами предприятия: бухгалтерским учетом, торговлей, производством, финансами, деятельностью персонала, работой складов и т. д. Отмечены недостатки данного программного решения: высокая стоимость и сложность внедрения.

Популярные решения для производства предоставляет компания IBM. Одно из них — IBM Watson — это суперкомпьютер фирмы IBM, оснащенный вопросно-ответной системой ис-

кусственного интеллекта, направленной на распознавание видов данных, взаимодействие с людьми на естественном языке, имеет возможность обучения. На его основе построено решение WatsonSupplyChain. Решение направлено на оптимизацию цепочек поставки. Для того чтобы удовлетворять растущие ожидания заказчиков, участники цепочки поставок должны отслеживать возможные сбои. Оптимизация цепочки поставок на основе искусственного интеллекта Watson помогает фирме прогнозировать, оценивать и предотвращать нарушения и риски при формировании и ежедневной работе сети бизнес-партнеров [13].

В статье [14] авторы делают акцент на цепочке поставок как важном этапе для производства. Эффективное и разумное управление поставками необходимо "для обеспечения доступности нужного продукта в нужное время и в нужном месте по подходящей цене". В статье основное внимание уделяется взаимоотношениям между предпринимателями и поставщиками услуг, работающими в парадигме совместной цифровой экономики с применением технологий блокчейн.

Другое популярное решение от IBM для производства, выполняющее функции управления ресурсами на предприятии, — Maximo [15]. Maximo в сочетании с данными интернета вещей, получаемыми от сотрудников, датчиков и устройств, выполняет оповещение о заканчиваемых ресурсах, что обеспечивает сокращение времени внеплановых простоев, повышение эффективности работы в целом.

Пример системы автоматизации бизнес-процессов в конкретной отрасли (в аграрном секторе) описан в работе [16]. Авторы утверждают, что для цифровизации в аграрном секторе создаваемая платформа должна иметь функции

обеспечения сквозной цифровизации всех физических активов и их интеграцию в цифровые экосистемы, предоставление и использование цифровых услуг, электронную обработку всех видов информации, поддержку информационного взаимодействия, использование бизнес-аналитики, управление технологическими процессами, усиление межотраслевого взаимодействия и вовлечение в предметно-ориентированные кластеры. Также в работе отмечается, что невозможно создать платформу без использования сторонних готовых решений. Таким образом, они приходят к утверждению, что цифровая экосистема — это социотехническая система, составленная из совокупности компьютерных программ с распределенным взаимодействием и взаимным использованием агентами в условиях эволюционного саморазвития.

Пример платформы для обмена инновационными разработками в промышленности с возможностью моделирования жизненного цикла промышленного производства, разработки и внедрения стратегий для развития процессов производства описан в работе [17].

В работе [18] описана технологическая платформа для легкой промышленности, которая ориентирована в первую очередь на модернизацию экономики. Развитие подобного проекта будет способствовать повышению эффективности сырьевых секторов, снижению энергоемкости российского производства.

На законодательном уровне в России Министерство экономического развития и торговли разработало целый ряд коммуникационных площадок [19], позволяющих взаимодействовать бизнесу, науке, потребителю и государству по вопросам модернизации и научно-технического развития по определенным технологическим направлениям. Пример популярного

Сравнение программных решений для цифровизации этапов и процессов деятельности фирмы

Задачи	Программные решения					
	Taobao	Amazon	IBM MAXITO	Watson SupplyChain	SAP	IC производство
Определение одного или нескольких мест размещения производств						
Планирование технологического процесса и выбор необходимых ресурсов для осуществления производства	+	+				+
Выбор оптимальных способов и мест приобретения ресурсов				+	+	+
Управление процессом поставок и закупок ресурсов и их оплатой						+
Производственная деятельность в рамках технологического процесса			+			+
Реализация готовой продукции и их оплата	+	+				

государственного информационного сервиса, интегрирующего различные системы и предусматривающего подсистему управления, является сервис государственных услуг [20].

На основе приведенного обзора литературы проведем сравнение программных решений с точки зрения решаемых задач в рамках выделенных нами выше этапов и процессов организации производства.

Таким образом, анализ имеющихся на рынке решений показал, что несмотря на широкий спектр российских и зарубежных платформ не все задачи, необходимые для процессов этапа организации и планирования производства, включающие выбор оптимального размещения, анализ спроса на продукцию, выбор надежных поставщиков и ряд других, покрываются современными программными решениями (см. таблицу). Именно поэтому разработка новых решений, использующих современные информационные технологии для поддержки различных этапов цифрового производства, является актуальной задачей, направленной на улучшение функционирования производства в экономике страны в целом.

2. Этапы организации цифрового производства

На основе анализа литературы, направлений деятельности и процессов организации деятельности фирмы сформулируем основные этапы организации цифрового производства:

- 1) оптимизация размещения (выбор места) производства;
- 2) выбор оптимального поставщика ресурсов;
- 3) оптимизация доставки и хранения ресурсов до места производства;
- 4) управление процессом поставок и закупок;
- 5) реализация защищенных контрактов, обязательств доставки продукции.

Любое производство, в том числе цифровое, начинается с выбора места его размещения. Необходимо определить желаемые регионы страны для размещения производства с учетом различных факторов, основными из которых являются спрос на продукцию в регионе, расстояние до поставщиков ресурсов и рынков сбыта. Эта проблема очень актуальна, в частности, для проектов комплексного развития и освоения территорий Дальнего Востока и Байкальского региона, которые относятся к категории слабоосвоенных территорий, одним из ключевых признаков которых является неразвитость транспортной инфраструктуры,

прежде всего, на подходах к портам, крупным промышленным районам, городским агломерациям и новым месторождениям полезных ископаемых и природных ресурсов в целом. Поэтому большое значение имеет предварительное создание таких информационно-вычислительных моделей, на основе которых возможно предварительное моделирование, анализ и оценка различных вариантов транспортно-логистических решений и решений о размещении производств до их применения в реальных условиях.

Следующим важным этапом как при организации производства, так и при его функционировании является поиск ресурсов для производства и выбор оптимального поставщика из множества возможных и оптимизация доставки продукции. В настоящее время можно выделить две основные проблемы данного этапа. Первая — отсутствие единой базы поставщиков, что приводит к необходимости их поиска в ручном, не автоматизированном режиме на одном из сайтов агрегаторов. И вторая — необходимость проверки надежности поставщика и его продукции, что также осуществляется в подобном ручном режиме.

Третий этап — оптимизация доставки и хранения ресурсов до места производства, где логистика имеет важное значение, так как это непосредственно влияет на издержки производства, от которых зависит себестоимость производимой продукции. Эффективность логистики состоит из времени доставки и ее стоимости, от этих двух критериев зависит эффективность производства в целом.

Четвертый этап — управление процессом поставок и закупок, задача данного этапа — поддерживать производство в актуальном состоянии. Необходимо отслеживать остатки ресурсов, заказывать новые, управлять логистикой, чтобы производство не останавливалось. Для бизнеса остановка производства из-за нехватки ресурсов является критически важной проблемой.

Пятым этапом является реализация защищенных контрактов, обязательств доставки продукции. После того как на втором этапе определены оптимальные поставщики, на третьем этапе определена оптимальная логистика, необходимо организовать процесс доставки ресурсов до места производства. Между поставщиком и предпринимателем заключается контракт, в котором оговариваются определенные условия. На этом этапе важно знать наиболее полную информацию о поставщике

для того, чтобы быть уверенным в легитимности его деятельности как юридического лица, его сделках и качестве продукции.

Описанные этапы организации производства могут быть реализованы как традиционными методами телефонных и личных переговоров, устных и письменных сообщений, передаваемых по телекоммуникационным каналам, так и с использованием цифровых сервисов. В настоящее время основным недостатком использования цифровых сервисов является их узкая специализация по решаемым задачам. Каждый сервис имеет свой интерфейс, подход, требования к знаниям сотрудников, что приводит к необходимости использования нескольких программных сервисов для выполнения этапа, привлечения большого числа сотрудников, что значительно влияет на усложнение внутрифирменной организации и управления. Объединение всех решений в единую систему позволит избежать этого недостатка. Такой подход обладает дополнительными преимуществами по сравнению с традиционными методами работы: наибольшей скоростью обмена информацией, протоколированием событий, автоматизацией рутинной работы.

Использование предлагаемой цифровой платформы позволит минимизировать издержки не только фирмы-получателя ресурсов, но и других фирм, связанных в единую производственно-технологическую цепочку: поставщиков ресурсов, компаний, осуществляющих доставку и хранение ресурсов.

Таким образом, разработка цифровой платформы, которая будет включать в себя вышеописанные этапы организации производства, позволит значительно упростить и уменьшить объем внутрифирменных затрат, что повысит конкурентоспособность подобных предприятий в данной отрасли и, как следствие, позволит увеличить число новых предприятий и производимой продукции, что положительно повлияет на экономику страны.

3. Основные требования к программной платформе

На основе анализа литературы, с учетом условий современной стадии развития технологий для организации производства можно выделить ключевые требования к функциональности и использованию программной платформы (ПП), состоящей из комплекса

взаимосвязанных программных средств для организации производства:

1. ПП должна являться облачным решением для обеспечения постоянной доступности пользователей и предоставлять доступ через веб-интерфейс к своей функциональности как предпринимателю, так и поставщику.

2. ПП должна обеспечивать удовлетворение интересов как предпринимателя, который планирует организовать производство, так и поставщика, который готов предоставить ресурсы для производства. Совмещение интересов участников в одной системе позволит избежать участия "третьих лиц".

3. ПП должна поддерживать работу с гетерогенными данными.

4. ПП должна предоставлять возможность сбыта продукции внутри системы для конкретных производств другим предпринимателям.

5. ПП должна быть расширяема, иметь модульную архитектуру в целях простой интеграции новых решений, направленных на проблемы с организацией производства.

6. ПП должна содержать систему рейтинга для поставщиков: рейтинг по числу выполненных контрактов доставки продукции, рейтинг качества.

7. ПП должна соблюдать свойства открытости (доступность, понятность, релевантность).

8. ПП должна поддерживать свойства систем: отказоустойчивость, распределенное хранение, неизменность данных.

9. ПП должна выполнять роль агрегатора, предоставляя данные платформы по API для других сервисов.

Ключевой идеей, которая лежит в основе построения инструментария, является автоматизация процесса организации производства и включение в этот процесс всех заинтересованных участников: предпринимателей и поставщиков.

4. Архитектура и функциональные модули

Основными архитектурными решениями, которые реализуют сформулированные выше требования, является создание децентрализованной платформы (как публичной сети), состоящей из независимых программных модулей (облачных сервисов), связанных в единую систему, число которых является расширяемым. Модуль осуществления смарт-контрактов реализуется с использованием блокчейн-технологии.

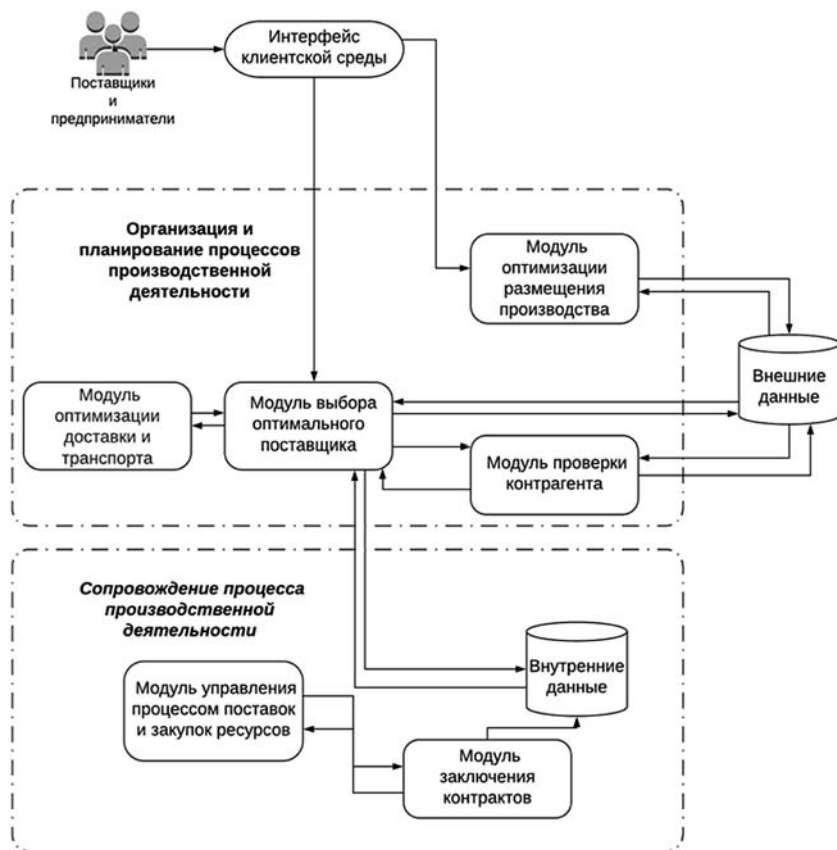


Рис. 1. Функциональные модули и их связи

Инструментальный комплекс состоит из блока организации и планирования процессов производственной деятельности и блока модулей, сопровождающих осуществление процесса производственной деятельности. Схема взаимодействия функциональных модулей (ФМ) представлена на рис. 1.

Блок *организации и планирования процессов производственной деятельности* включает следующие модули:

- *модуль оптимизации размещения производства*. Данный модуль необходим на первичном этапе планирования производства. Он обеспечивает проведение анализа и оценку различных вариантов размещения с учетом как транспортно-логистических решений при доставке продуктов, требуемых для производства, так и наличия спроса на производимый продукт и его доставку в регион(ы) с наибольшим спросом;
- *модуль выбора оптимального поставщика* предоставляет поставщикам и производителям средства интеллектуального поиска друг друга, учитывая цены, стоимость и время доставки, репутацию, существующие деловые связи, а также цепочки контрактов

связанных между собой товаров. Система полностью рассматривается как единая сеть экономических операций;

- *модуль оптимизации доставки и транспорта* используется для решения задачи оптимизации поставки/производства ресурсов в сети пространственно распределенных экономических агентов с учетом различных видов транспорта;
- *модуль проверки контрагента* выполняет проверку по таким критериям, как данные о поставщике, данные о выполнении его обязательств перед другими предпринимателями, уплата налогов, доля владения и др.;
- *интерфейс клиентской среды* выступает основным средством взаимодействия предпринимателей и поставщиков с платформой, предоставляющей решения для производства. С помощью этого компонента осуществляется взаимодействие с модулями внутри платформы.

Блок *сопровождения процесса производственной деятельности* включает следующие модули:

- *модуль управления процессом поставок и закупок ресурсов* предназначен для создания цикла производства, включая возможность заказывать продукцию в определенные этапы производства. Этот модуль позволяет отслеживать остатки ресурсов на предприятии и на основе анализа заказывать необходимые ресурсы, чтобы избежать простоя в работе на предприятии. Интеллектуальная система модуля генерирует рекомендации, помогая пользователю поддерживать процессы производства;
- *модуль заключения контрактов* включает в себя платежную систему на основе квазивалюты. Квазивалюта используется для полного управления денежными средствами внутри системы с помощью умных контрактов. Это позволяет реализовать механизм заморозки средств, необходимый системе при выполнении обмена ресурсами между поставщиком и предпринимателем, для обеспечения наибольшей надежности. Необходимо также обеспечить их конвертацию в традиционные виды валют при входе и выходе средств из системы. В основе

модуля лежит эскроу-механизм — это схема работы, при которой две стороны поручают третьей хранить финансовый инструмент или актив на время совершения операции, пока осуществляется доставка ресурса. При использовании этого подхода третья сторона выступает в качестве гаранта, которая контролирует весь процесс и убеждается в исполнении сторонами своих обязательств.

Программная реализация блока *организации и планирования процессов производственной деятельности* основана на использовании облачной архитектуры для обеспечения постоянной доступности. ФМ в данном контексте

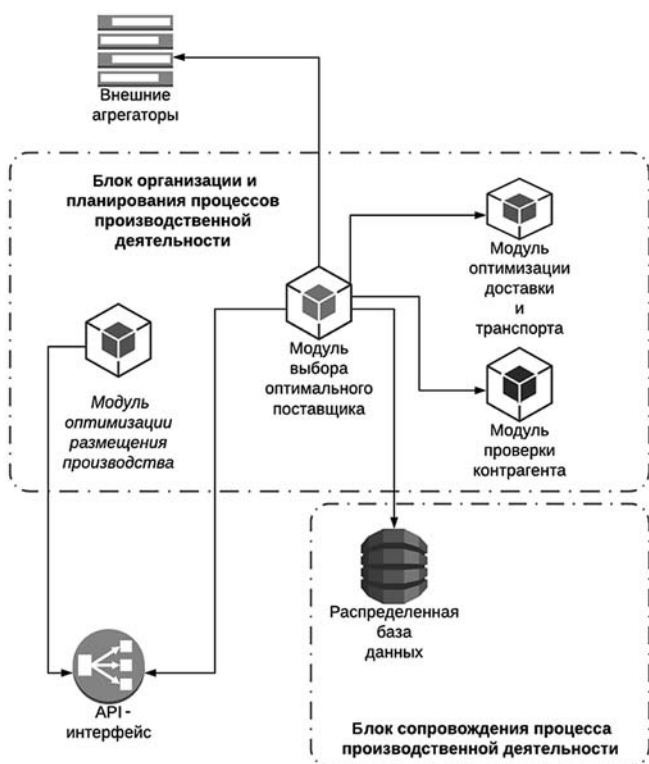


Рис. 2. Программная реализация блока организации и планирования процессов производственной деятельности

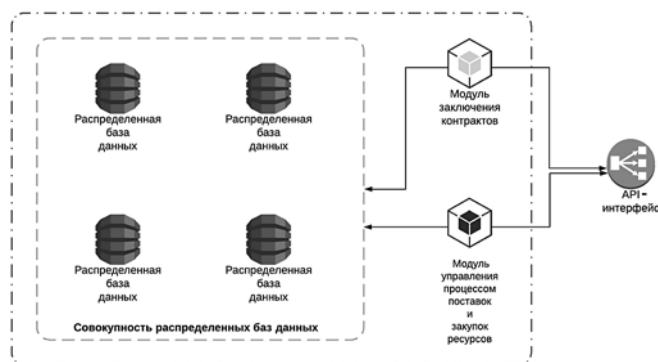


Рис. 3. Программная реализация блока сопровождения процесса производственной деятельности

сте выступают как независимые приложения (рис. 2), которые функционируют как внутри платформы, так и предоставляя API-интерфейс для различных внешних интеграций. Каждый ФМ использует собственную структуру данных для входных и выходных параметров.

Программная реализация блока *сопровождения процесса производственной деятельности* имеет децентрализованную архитектуру с распределенной базой данных и предоставляет API доступ для работы с данными, хранящимися в ней. Алгоритм консенсуса используется для добавления новых данных. Здесь используются решения, основанные на технологии блокчейн для хранения информации в неизменном виде, для таких сущностей, как отзывы, рейтинги поставщиков и "умной" логики, которая используется для заключения контрактов (рис. 3).

Заключение

В работе были рассмотрены направления развития цифрового производства, показаны этапы, которые необходимо выполнить при организации производства. Для каждого этапа определены функциональные модули, из которых состоит платформа. Показаны взаимосвязи между модулями. В настоящее время уже разработаны и экспериментально исследованы ряд модулей системы [21, 22], ведется работа по реализации полнофункционального инструментального комплекса [23].

Список литературы

1. **Официальный сайт IBM:** Производственные решения. URL: <https://www.ibm.com/industries/ru-ru/manufacturing/solutions.html> (дата обращения: 19.09.2018).
2. **Официальный сайт IBM:** Список отраслей. URL: <https://www.ibm.com/industries> (дата обращения: 19.09.2018).
3. **Organization and management of the production process of the company** // Science and Higher Education in Function of Sustainable Development. URL: <http://www.vpts.edu.rs/sed17/CD%20Proceedings%202017/proceedings/7-2.pdf> (дата обращения: 19.09.2018).
4. **Сайт Smetoolkit:** 5 этапов планирования производства. URL: <http://www.smetoolkit.org/en/content/production-planning-5-steps> (дата обращения: 10.11.2018).
5. **Официальный сайт Сколково.** Цифровые платформы и экосистемы финансовой инклюзивности. Российский опыт. URL: https://iems.skolkovo.ru/downloads/documents/SKOLKOVO_IEMS/Research_Reports/SKOLKOVO_IEMS_Research_2015-11-11_ru.pdf (дата обращения: 10.04.2018).
6. **Карцхиа А. А.** Цифровая революция: новые технологии и новая реальность // Правовая информатика. 2017. № 1. С. 13–18.
7. **Christidis К.** Blockchains and Smart Contracts for the Internet of Things // IEEE. 2016. Vol. 4, N. 11. P. 2292–2303.

8. **Кузьмина Е. И.** Феномен эффективности компании alibaba group Джека Ма // Электронный журнал nauka-gastudent.ru. 2016. № 9. С. 033.

9. **Назаров В. А.** Развитие новых информационных технологий, обеспечивающих эффективное функционирование электронного бизнеса // Экономика и социум: современные модели развития. 2017. № 15. С. 54–66.

10. **Комплекс программ 1С: Обзор решений 1С.** URL: <http://c-school.ru/education-1-enterprise.html> (дата обращения: 10.11.2018).

11. **Функции 1С: Функции 1С для управления фирмой.** URL: <http://v8.1c.ru/small.biz/usef/> (дата обращения: 10.11.2018).

12. **Семухин С. О.** Характеристика логистической системы "SAP" // Концепт. 2016. № 2. С. 12–15.

13. **Оптимизации работы поставки от IBM: Оптимизация поставки.** URL: <https://www.ibm.com/watson/supply-chain/resources/visibility-made-easy/> (дата обращения: 10.11.2018).

14. **Куприяновский В. П., Снягов С. А.** Цифровые цепи поставок и технологии на базе блокчейн в совместной экономике // International Journal of Open Information Technologies. 2017. Vol. 5, N. 8. С. 12–15.

15. **Обзор IBM Maximo [Электронный ресурс]: IBM-Maximo: сайт.** URL: <https://www.ibm.com/ru-ru/marketplace/maximo> (дата обращения: 10.11.2018).

16. **Алетдинова А. А.** Инновационное развитие аграрного сектора на основе цифровизации и создания технологиче-

ских платформ // Иннов. электронный научный журнал. 2017. № 4. С. 33.

17. **Корягин С. И.** Цифровая трансформация промышленности: теория и практика // Baltic Federal University of Immanuel Kant. 2016. С. 247–267.

18. **Шинкевич А. И.** Перспективы развития российских технологических платформ в легкой промышленности // Теория и практика сервиса: экономика, социальная сфера, технологии. 2016. С. 348–349.

19. **Фундаментальные исследования.** Научный журнал // Приоритетные платформы. URL: <https://www.fundamental-research.ru/ru/article/view?id=33392> (дата обращения: 10.04.2018)

20. **Официальный сайт государственных услуг России: Информационный портал государственных услуг России.** URL: <https://www.gosuslugi.ru>.

21. **Velichko A. S., Gribova V. V., Fedorishchev L. A.** Simulation software for multicommodity flows model of interregional trade // IEEE Xplore. 2018. DOI: 10.1109/RPC.2018.8482140.

22. **Velichko A. S., Gribova V. V., Fedorishchev L. A.** Software System for Interactive Simulation of Interregional Trade // CEUR Workshop Proceedings. 2016. Vol. 1623. P. 383–393.

23. **Величко А. С., Грибова В. В., Федорищев Л. А.** Облачный сервис для интерактивного моделирования межтерриториальной торговли // Моделирование и анализ информационных систем. 2016. Т. 23, № 4. С. 412–426.

V. V. Gribova, Research Deputy Director, e-mail: gribova@iacp.dvo.ru,

A. S. Velichko, Researcher, e-mail: vandre@iacp.dvo.ru,

A. V. Kolmogorov, PhD Student, e-mail: sunrise3323@gmail.com,

Institute of Automation and Control Processes Far Eastern Branch of the Russian Academy of Sciences

Digital Platform Concept for Business Activities

The paper deals with the problem of organizing digital production. The survey of the existing software solutions, their advantages and disadvantages are analyzed. On the basis of the survey, the main stages of the organization of digital production are identified. They are: optimization of placement (choice of place) of production; selection of the optimal supplier of resources; optimization of delivery and storage of resources to the place of production; management of the process of supply and procurement; implementation of protected contracts, product delivery obligations. Based on the analysis of the literature, taking into account the conditions of the modern stage of development of technologies for the organization of production, we can identify the key requirements for the functionality and use of the software platform, consisting of a set of interrelated software for the organization of production. The main architectural solutions are the creation of a decentralized platform (as a public network), consisting of independent software modules (cloud services), connected into a tool with expandable number of modules. The module of smart contracts is implemented using blockchain technology. The tool consists of the block of the organization and planning of processes of production activity and the block of the modules accompanying implementation of process of production activity. Currently, a number of modules of the system have been developed and experimentally tested, now we are developing the full-featured tool.

Keywords: digital economy, organization of production and transportation, information platform, instrumental tools complex

DOI: 10.17587/it.25.502-511

Acknowledgment: The work is carried out with the partial financial support of the RFBR (projects 18-07-01079, 19-07-00244).

References

1. **Official** website IBM: Manufacturing Solutions, available at: <https://www.ibm.com/industries/ru-ru/manufacturing/solutions.html> (date of access: 19.09.2018) (in Russian).
2. **Official** website IBM: Industries, available at: <https://www.ibm.com/industries> (date of access: 19.09.2018) (in Russian).
3. **Organization** and management of the production process of the company. Science and Higher Education in Function of Sustainable Development, available at: <http://www.vpts.edu.rs/sed17/CD%20Proceedings%202017/proceedings/7-2.pdf> (date of access: 19.09.2018).
4. **Website** Smetoolkit: Production planning — 5 steps, available at: <http://www.smetoolkit.org/en/content/production-planning-5-steps> (date of access: 10.11.2018) (in Russian).
5. **Official** website Skolkovo: Cifrovyje platformy i jekosistemy finansovoj inkluzivnosti. Rossijskij opyt (Digital platforms and financial inclusion ecosystems. Russian experience), available at: [https://iems.skolkovo.ru/downloads/documents/SKOLKOVO_IEMS_Research_2015-11-11_ru.pdf](https://iems.skolkovo.ru/downloads/documents/SKOLKOVO_IEMS_Research_Reports/SKOLKOVO_IEMS_Research_2015-11-11_ru.pdf) (date of access: 10.04.2018) (in Russian).
6. **Karchia A. A.** The digital revolution: new technologies and new reality, *Pravovaja Informatika*, 2017, no. 1, pp. 13–18 (in Russian).
7. **Christidis K.** Blockchains and Smart Contracts for the Internet of Things, *IEEE*, 2016, vol. 4, no. 11, pp. 2292–2303.
8. **Kuz'mina E. I.** The phenomenon of the effectiveness of the company alibaba group Jack Ma, *Jelektronnyj zhurnal nauka-rastudent.ru*, 2016, no. 9, pp. 033 (in Russian).
9. **Nazarov V. A.** The development of new information technologies to ensure the effective functioning of e-business, *Jekonomika i socium: sovremennye modeli razvitija*, 2017, no. 15, pp. 54–66 (in Russian).
10. **Review** of IS company solutions, available at: <http://c-school.ru/education-1-enterprise.html> (date of access: 10.11.2018) (in Russian).
11. **Functions** of IS software to manage the company, available at: <http://v8.lc.ru/small.biz/usef/> (date of access: 10.11.2018) (in Russian).
12. **Semuhin S. O.** Characteristics of the SAP logistics system, *Koncept*, 2016, no. 2, pp. 12–15 (in Russian).
13. **Supply** chain optimization, available at: <https://www.ibm.com/watson/supply-chain/resources/visibility-made-easy> (date of access: 10.11.2018) (in Russian).
14. **Kuprijanovskij V. P., Sinjagov S. A.** Digital supply chains and blockchain-based technologies in a collaborative economy, *International Journal of Open Information Technologies*, 2017, vol. 5, no. 8, pp. 12–15 (in Russian).
15. **IBM** Maximo review, available at: <https://www.ibm.com/ru-ru/marketplace/maximo> (date of access: 10.11.2018) (in Russian).
16. **Aletdinova A. A.** Innovative development of the agricultural sector on the basis of digitalization and the creation of technology platforms, *Innov. jelektronnyj nauchnyj zhurnal*, 2017, no. 4, pp. 33. (in Russian).
17. **Korjagin S. I.** Digital Industry Transformation: Theory and Practice, Baltic Federal University of Immanuel Kant, 2016, pp. 247–267 (in Russian).
18. **Shinkevich A. I.** Prospects for the development of Russian technological platforms in the light industry, *Teorija i praktika servisa: jekonomika, social'naja sfera, tehnologii*, 2016, pp. 348–349 (in Russian).
19. **Priority** platforms, *Fundamental'nye issledovanija*, available at: <https://www.fundamental-research.ru/ru/article/view?id=33392> (date of access: 10.04.2018) (in Russian).
20. **Official** website. Information portal of public services in Russia, available at: <https://www.gosuslugi.ru> (in Russian).
21. **Velichko A. S., Gribova V. V., Fedorishchev L. A.** Simulation software for multicommodity flows model of interregional trade, *IEEE Xplore*, 2018, DOI: 10.1109/RPC.2018.8482140.
22. **Velichko A. S., Gribova V. V., Fedorishchev L. A.** Software System for Interactive Simulation of Interregional Trade, *CEUR Workshop Proceedings*, 2016, vol. 1623, pp. 383–393.
23. **Velichko A. S., Gribova V. V., Fedorishchev L. A.** Cloud service for interactive simulation of interregional trade, *Modelirovanie i analiz informacionnyh system*, 2016, vol. 23, no. 4, pp. 412–426 (in Russian).

Адрес редакции:

107076, Москва, Стромьинский пер., 4

Телефон редакции журнала (499) 269-5510

E-mail: it@novtex.ru

Технический редактор *Е. В. Конова*.

Корректор *Е. В. Комиссарова*.

Сдано в набор 07.06.2019. Подписано в печать 26.07.2019. Формат 60×88 1/8. Бумага офсетная.

Усл. печ. л. 8,86. Заказ ИТ819. Цена договорная.

Журнал зарегистрирован в Министерстве Российской Федерации по делам печати, телерадиовещания и средств массовых коммуникаций.

Свидетельство о регистрации ПИ № 77-15565 от 02 июня 2003 г.

Оригинал-макет ООО "Авансед солюшнз". Отпечатано в ООО "Авансед солюшнз".

119071, г. Москва, Ленинский пр-т, д. 19, стр. 1. Сайт: www.aov.ru

IV Международная научно-практическая конференция

РАДИОИНФОКОМ-2019

состоится 11 – 15 ноября 2019 г. в РТУ МИРЭА

Минобрнауки России, Федеральное государственное бюджетное образовательное учреждение высшего образования МИРЭА – Российский технологический университет приглашает Вас принять участие в IV Международной научно-практической конференции «Актуальные проблемы и перспективы развития радиотехнических и инфокоммуникационных систем» («РАДИОИНФОКОМ-2019»), которая состоится 11 – 15 ноября 2019 года в РТУ МИРЭА по адресу: 119454, г. Москва, проспект Вернадского, д.78.

ЦЕЛЬ КОНФЕРЕНЦИИ

обсуждение проблем, актуальных направлений и перспектив развития одного из важнейших направлений мировой экономики – радиоэлектронной и телекоммуникационной индустрии, обеспечивающей функционирование жизненно важных элементов и показателей государственной инфраструктуры гражданского и оборонного назначения.

СЕКЦИИ КОНФЕРЕНЦИИ

1. Телекоммуникации и радиотехника
2. Радиоэлектронные системы и комплексы
3. Обработка и передача информации по каналам радиосвязи
4. Радиофизика и электродинамика
5. Радиоинжиниринг
6. Проектирование и технология радиоэлектронных средств
7. Инженерная экология
8. Подготовка профессиональных кадров в инновационном университетском кластере
9. Обеспечение качества электронных средств, стандартизация и сертификация

В рамках конференции будет учрежден круглый стол, посвященный обсуждению перспективных направлений, проектов и научно-практических разработок в области инновационного развития цифровой экономики, а также традиционно проведен конкурс лучших докладов молодых ученых.

Материалы докладов конференции будут опубликованы в электронном сборнике научных трудов с размещением в базе РИНЦ.



Узнать подробную информацию о конференции или обратиться с вопросом, оставив сообщение в форме заявки, можно на сайте конференции <http://forum.mirea.ru/>

Рисунок к статье В. Е. Гвоздева, О. И. Христодуло

«ИНФОРМАЦИОННАЯ ПОДДЕРЖКА ПРИНЯТИЯ РЕШЕНИЙ В ЗАДАЧАХ УПРАВЛЕНИЯ ОТХОДАМИ НА ОСНОВЕ МАТЕМАТИКО-ГЕОИНФОРМАЦИОННОГО МОДЕЛИРОВАНИЯ»

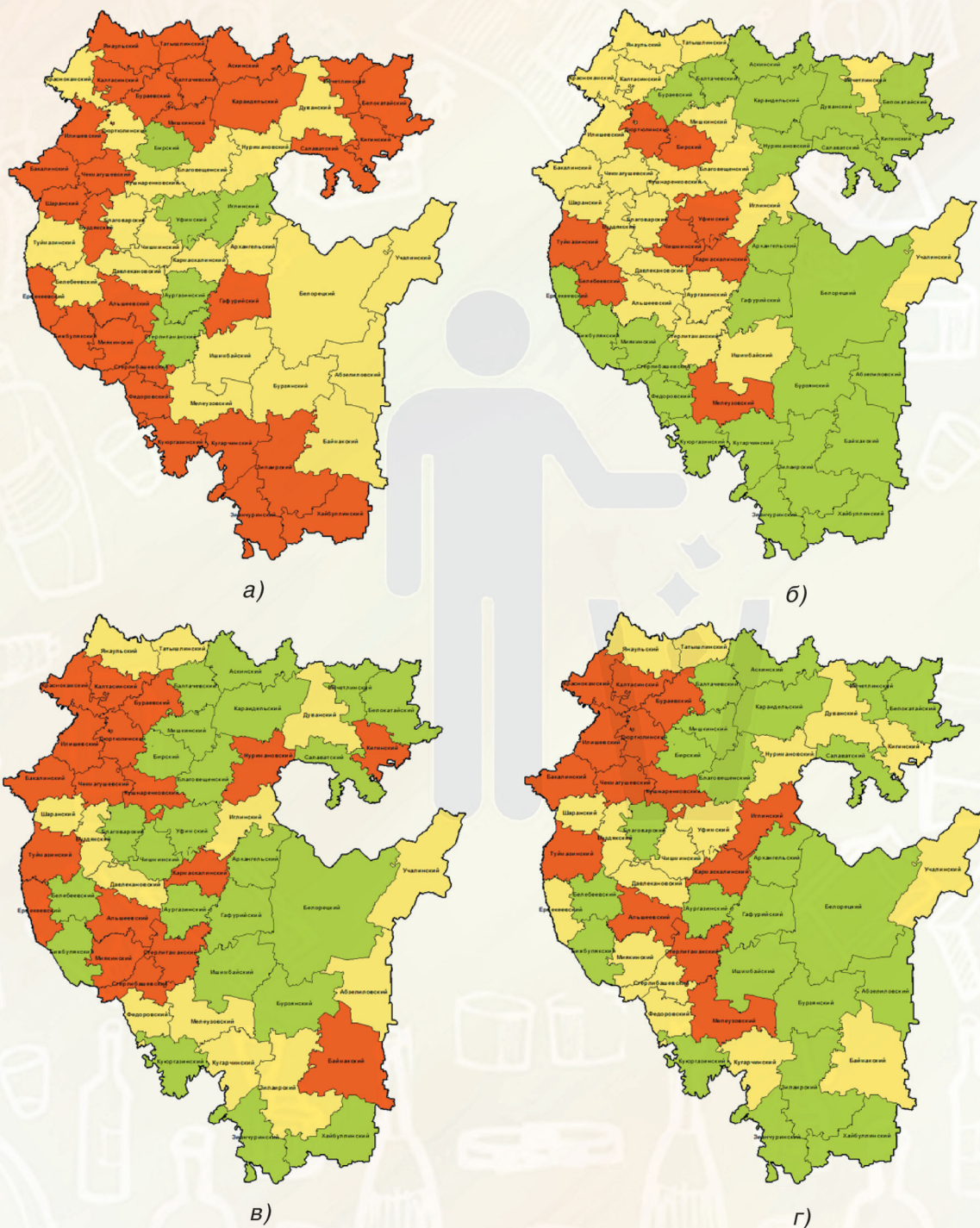


Рис. 2. Количество отходов на территории РБ:

а – количество образованных отходов на душу населения; б – количество образованных отходов на 1 км²; в – количество использованных и обезвреженных отходов на душу населения; г – количество использованных и обезвреженных отходов на 1 км²;
■ – низкий уровень негативного воздействия; ■ – средний уровень негативного воздействия; ■ – высокий уровень негативного воздействия

Издательство «НОВЫЕ ТЕХНОЛОГИИ» выпускает научно-технические журналы



Ежемесячный теоретический и прикладной научно-технический журнал **ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

В журнале освещаются современное состояние, тенденции и перспективы развития основных направлений в области разработки, производства и применения информационных технологий.

Подписной индекс по Объединенному каталогу
«Пресса России» – 72656



Научно-практический
и учебно-методический журнал

БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

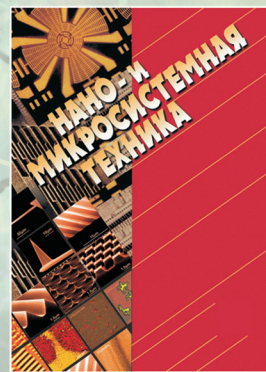
В журнале освещаются достижения и перспективы в области исследований, обеспечения и совершенствования защиты человека от всех видов опасностей производственной и природной среды, их контроля, мониторинга, предотвращения, ликвидации последствий аварий и катастроф, образования в сфере безопасности жизнедеятельности.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79963

Ежемесячный
междисциплинарный
теоретический и прикладной
научно-технический журнал

НАНО- и МИКРОСИСТЕМНАЯ ТЕХНИКА

В журнале освещаются современное состояние, тенденции и перспективы развития нано- и микросистемной техники, рассматриваются вопросы разработки и внедрения нано микросистем в различные области науки, технологии и производства.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 79493



Ежемесячный теоретический
и прикладной
научно-технический журнал

МЕХАТРОНИКА, АВТОМАТИЗАЦИЯ, УПРАВЛЕНИЕ

В журнале освещаются достижения в области мехатроники, интегрирующей механику, электронику, автоматику и информатику в целях совершенствования технологий производства и создания техники новых поколений. Рассматриваются актуальные проблемы теории и практики автоматического и автоматизированного управления техническими объектами и технологическими процессами в промышленности, энергетике и на транспорте.

Подписной индекс по
Объединенному каталогу
«Пресса России» – 79492

Теоретический
и прикладной
научно-технический журнал

ПРОГРАММНАЯ ИНЖЕНЕРИЯ

В журнале освещаются состояние и тенденции развития основных направлений индустрии программного обеспечения, связанных с проектированием, конструированием, архитектурой, обеспечением качества и сопровождением жизненного цикла программного обеспечения, а также рассматриваются достижения в области создания и эксплуатации прикладных программно-информационных систем во всех областях человеческой деятельности.



Подписной индекс по
Объединенному каталогу
«Пресса России» – 22765

Адрес редакции журналов для авторов и подписчиков:

107076, Москва, Стромьинский пер., 4. Издательство "НОВЫЕ ТЕХНОЛОГИИ".
Тел.: (499) 269-55-10, 269-53-97. Факс: (499) 269-55-10. E-mail: antonov@novtex.ru